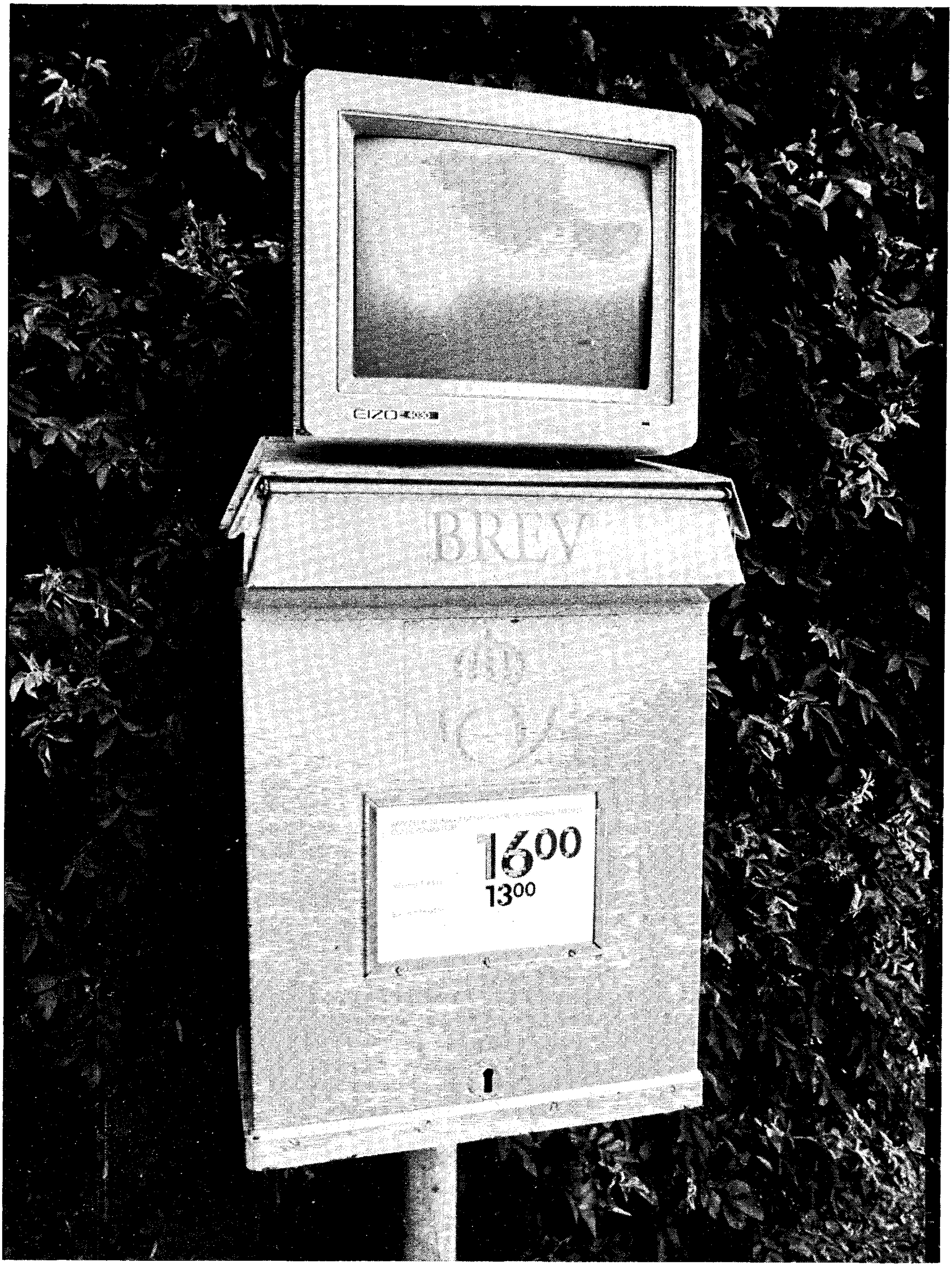
ABC BLADET

1993 nummer 3

ABC-KLUBBENS MEDLEMSBLAD FÖR BLANDAD INFORMATION TILL BÅDE NYTTA OCH NÖJE



Hur datorn lurar oss eller Mera om diofantiska egyptiska ekvationer

Problemet med kamelerna än en gång

I förra omgången efterfrågades heltalslösningar till ekvationen:

1/a + 1/b + 1/c + 1/d = 1 (1)

Utgångspunkten var den gamle beduninens testamente. Hans tre söner skulle fördela 17 kameler i delarna 1/2, 1/3 och 1/9. Dessa bråk blir tillsammans 17/18 och man kan alltså säga att talföljden (2,6,9,18) utgör en lösning:

1/2 +1/6 + 1/9 + 1/18 = 1 (2)

Det frågades efter fler lösningar, alltså om det fanns andra kamelmängder som kunde fördelas samma metod: "lägg till en, gör fördelningen och få en över".

Att skriva ett tämligen enkelt program i valfritt språk för att leta reda på tal som uppfyller villkoret är inte svårt. Ett exempel i basic2 visas i detta nummer med övliga kommentarer.

Det visar sig finnas 14 lösningar till ekvationen (1), men två av dem duger inte för kamelproblemet:

2, 3, 10, 15 (3)

och

3, 4, 4, 6 (4)

Översatt till kameler skulle (3) betyda att man skulle fördela 15 kameler i en halva, en tredjedel och en tiondel och få en över. Det fungerar inte på levande kameler, eftersom 15/10 inte är någon heltalslösning.

På motsvarande vis skulle man i (4) behöva dela 6 kameler med 4, vilket inte heller går jämnt upp.

Även om det inte finns någon heltalslösning på kamelproblemet med dessa värden, är det ändå korrekta svar på den ursprungliga ekvationen (1)!

Det räcker alltså inte med att få fram lösningarna till ekvationen. Man måste också kolla lösningarna mot det ursprungliga problemet. Eller uttryckt på ett annat sätt: Den matematiska modellen (1) är inte bra nog för att ge endast korrekta lösningar till kamelproblemet.

I just det här problemet kan man t ex lägga till villkoret att d måste vara jämnt delbart med vardera a, b och c. Men det kommer man nog på först i efterhand.

Datorn lurar oss!

För oss vanliga hackare är det ganska välkänt att våra program inte alltid gör det vi hade tänkt oss, trots att de gör precis det vi befallt. När det gäller ett enkelt problem, som lätt kan kontrolleras för hand, eller ett litet antal lösnigar som man tämligen lätt kan överblicka, gör det inte så mycket att datorn är litet nyckfull. Men hur blir det med mycket stora problem, där resultatet är svårt att i detalj kontrollera?

Jag har i många artiklar, inte minst här i ABC-bladet, tjatat om behovet att vara misstänksam och kolla, kolla och kolla om igen innan man accepterar de resultat en dator spottar ur sig.

Ingen riksdagsman kunde väl ens drömma om att det skattebeslut han/hon var med om att fatta hösten 1991 skulle leda till högkonjunktur för ett bananförsäljande företag i stockholmstrakten och till att alla riksdagsmän och tjänstemän i riksdagen skulle synas i korridorerna bärande på bananer! (Se artiklar i ABC-bladet, bl a 92:3 s 7).

Vad är det mer vi inte anar?

Ännu mera lurt!

Men tillbaka till problemet. Det egendomliga är att lösningen (3) inte kom fram med mitt program. Trots att SumC=1/2+1/3+1/10 helt korrekt beräknades till 0,833333 och 1/15 till 6,66666E-2. Tillsammans borde detta, med korrekt avrundning bli 1, och så angavs det av programmet vid anfordran. Men icke desto mindre ratades värdet vid kontrollen IF Sum=1 - trots att ;Sum=1 av programmet, på begäran, förklarades som 1 dvs sant.

Man måste förstås fråga sig varför? Jag har inte orkat och klarat att tränga så djupt in i problemet att jag kan garantera ett korrekt svar, men jag gissar att det har med datoraritmetiken att göra.

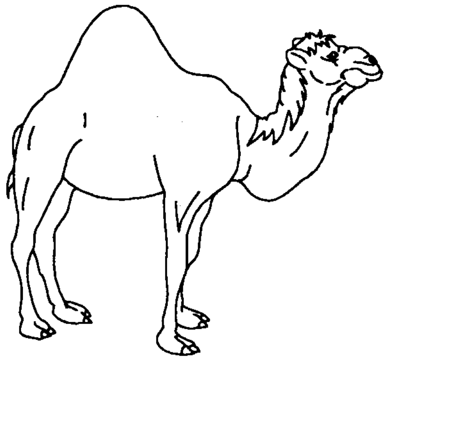
Allmänna bråk som 1/15 kan aldrig anges exakt som decimalbråk. Datorn räknar med motsvarande bråk inom det binära talsystemet (binimalbråk?), där steget från en kolumn till nästa avgörs av faktorn 2 i stället för av 10 som i decimalbråk. Det blir alltså ett grövre mått, och troligen grövre avvikelser från det korrekta värdet. Kanske avvikelsen blir så stor att det inte sker korrekt avrundning till 1 så länge datorn arbetar internt med binära tal? Det kan vi inte se, när vi begär besked i decimala termer, för då blir det korrekt. (Detta behandlades utförligt i ABC-bladet 92:1 sid 3 Den bedrägliga nollan.)

Färdiglurat?

Då uppkommer nästa obehagliga fråga: Finns det fler värden som inte kommer att redovisas? En manuell kontroll av tresönersproblemet visar att så inte är fallet. Men om man ökar till fyra söner och söker motsvarande talföljder? Eller om man generaliserar problemet och vill beräkna någonting riktigt stort?

Det är nog klokt att fortsätta med en viss skepsis inför den oheliga alliansen mellan datorgenererade utsagor, politikers beslut och allmänhetens lättsinne.

<1384>
Sven Wickberg



ABC BLADET

Medlemsorgan för

ABC-klubben

Box 141 43
161 14 Bromma
ISSN 0349-3652

Redaktör o ansvarig utgivare
Ulf Sjöstrand
I redaktionskommitten
Claes Schibler o Sven Wickberg

ABC-klubbens postgirokonton

Medlemsavgifter:	15 33 36-3
Publikationer:	62 93 00-5
SuperKOM:	43 51 74-8
Bankgiro:	216-25 43

Telefoner:

08-80 17 25	"Prattelefon" till klubblokalen och kansliet. Kansliet träffas säkrast tisdagskvällar
08-80 15 22	Telefax

Monitorer:

08-80 64 40	Gruppnummer för allt upp till 2400
08-80 15 23	Gruppnummer för 9600, 14 400 och HST

1993 nummer 1
innehåll

Omslag: Elektronisk post? Foto: Lars Strömberg
Hur datorn lurar oss eller Mera om diofantiska egyptiska ekvationer av Sven Wickberg Oms 2
Redaktionssidan 1
Ledare 2
Hur skickar jag brev via InterNet och FidoNet från ABC-klubben och hur skickar andra brev till mig? av Lars Strömberg och Bo Kullmar 3
Trollkarlens lärling av Sven Wickberg 6
Ett förslag till lösning av problemt med kamelerna av Sven Wickberg 7
Vanliga frågor om teckensnitt 8
Ytterligare en omgång "Vanliga frågor om teckensnitt" 23
Kort lektion i PostScript av Bo Kullmar 27
Något om rutiner i BASIC.II av Jan Sundström 30
"You are not expexted to understand this" 31
Min elektroniska lekstuga av Bill Leksén 32
Linux och Linus Torvalds av Jan Smith Oms 3

Tryck: Svenskt Tryck AB i en upplaga om 2000 ex
Lämnad till tryck den 25 juli 1993

Ändrad produktionsplan för 1993

nr	manus- stopp	annons- bokningsstopp	till tryckeriet	till medlemmen
935	9/8	9/8	23/8	w937
936	13/9	13/9	27/9	w941
937	11/10	11/10	25/10	w945
938	8/11	8/11	22/11	w950

Denna produktionsplan är preliminär. Ändringar kan komma att ske under hösten.

Annonspriser

1/1-sida	185*260 mm	1750:-
1/2-sida	185*128 mm eller 90*260 mm	1050:-
1/3-sida	185* 85 mm eller 60*260 mm	700:-
1/4-sida	90*128 mm	650:-
1/6-sida	60*128 mm	450:-

Tillägg för begärd placering 15%
Radannonser för medlemmar utan särskild debitering

ABC-klubbens styrelse för 1993
(enligt årsmötet 1993-03-13)

Ordförande:	Bo Kullmar
Vice ordförande:	Jaan Tombach
Ledamöter:	Jan Smith
	Kjell Brealt
	Martin Lundberg
	Tom Sjöberg
	Ulf Sjöstrand
	Conny Westh
Suppleanter:	Henry Gessau
	Ulf Hedlund
	Carl Kövamees
	Mats Lillnor
	Benny Löfgren
	Per Sten
	Lars Strömberg

Medlemsavgifter 1993

Seniorer 225 SEK och juniorer 150 SEK.
Junior räknas man t o m det år man fyller 18 år. Ange därför personnumret när Du betalar in medlemsavgiften. Medlemsskapet är personligt och avser fysisk person. Medlemskapet räknas per kalenderår och Du får automatiskt det löpande årets förmåner retroaktivt om Du inte markerar annat årtal på talongen när medlemsavgiften betalas in.

Ledare

Sommar, det är den tid som vi i Sverige stänger butiken och ger ledigt åt alla. Detta uttalande kan ju se oskyldigt och ett uttalande som många ställer upp på. Men vad står det egentligen för?

Sommar är enligt en definition den tid som dagsmedeltemperaturen överstiger 10 grader Celsius. Sommar, för säkerhets skull dubblerat, Sommar, sommar, är enligt Sveriges radio ett program där aktuella personer får fritt framföra sina åsikter och det som ligger dem nära samt spela sin favoritmusik. I båda dessa fall verkar definitionen tagit efter kronan, de flyter. Numer menar man även att nederbörden skall avhålla sig från att falla, åtminstoner på den plats där just jag har valt att uppehålla mig. Beträffande radioprogrammet har man i stället flyttat på väderleksrapporten för att kunna erbjuda en service som börjar vid en jämn timme samtidigt som man har halverat omfånget.

Stänger butiken

En sanning med viss modifikation. Av de över 10 procenten som inte har någon butik att stänga skulle nog tycka att det vore trevligt att ha någon butik att stänga.

Något som har sagt många gånger i andra sammanhang men gäller även oss. Vi har inte en bättre tidning än våra medlemmar och deras bidrag tillåter oss av vara. Så därför skriv och berätta om Din verklighet i en värld som datoriseras. Din egna erfarenheter, funderingar och förtretligheter.

ABC-klubbens arbetssätt

Det har kommit en del synpunkter på artikeln av den nye medlemmen i förra nummret. För att bemöta de synpunkter som framförs där kan bara framföras att ABC-klubben är en ideel organisation som har funnits i drygt tio år. Medlemsantalet är cirka två tusen och medlemmarna spridda över hela landet. Styrelsen försöker fånga upp de önskemål från medlemmarna och genomföra dessa efter bästa förmåga, dels personellt och dels ekonomiskt.

Det finns dessutom två lokalavdelninger, en i Stockholm och en i Linköping, som försöker komplettera riksföreningens övergripande verksamhet med att ordna "lokal" verksamhet, med träffar och möten samt möjlighet att utnyttja gememsam utrustning. Styrelsen har hela tiden framhållit att det måste finnas ett lokalt intresse att en sådan verksamhet kommer igång. Det kommer inte att fungera särskilt länge eller på ett bra sätt om det är ett centralt "påbud" att en lokalavdelning skall dras igång

Styrelsen har däremot under föreningens hela tillvaro uppmuntrat tillkomsten av sådana lokala verksamheter. Den har tyvär inte varit särskilt framgångsrik med denna uppgift.

Föreliggande nummer

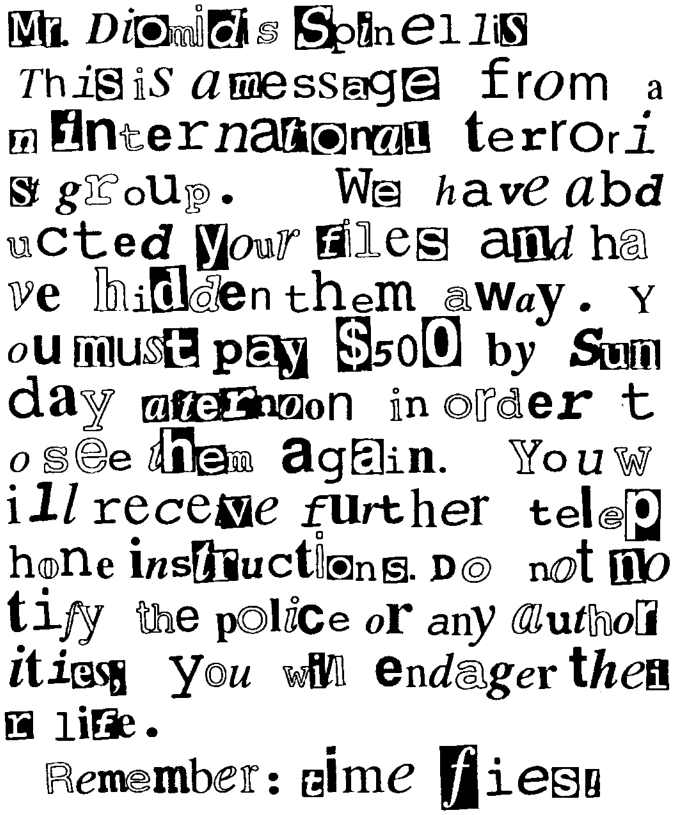
Varför ser det ut som det gör?. Att det är tunnt och kommer ut nu berörde jag redan i föregående mummer. Vidare undersökningar har givit vid handen att vi praktiskt sett kan gå tillbaka med fyra nummer per år och den tjockare varianten. Vi överväger också konsekvenserna av att komma ut med sex nummer per år. Detta beror på många faktorer. En av de viktigaste är ju faktiskt att ABC-bladet har tillräckligt med material som medlemmarna tycker är intressant och har lust att läsa.

Denna gången har vi inte några MSG-utdrag. Om dessa tas för ofta kommer de längre diskussionerna att bli något rumphuggna med den kortare mängden att välja ut ifrån. Vidare är det ett stressigt jobb att göra dessa utdrag, framför allt om det skall vara någorlunda fräscht material i dem.

Artiklarna

Sven W fortsätter och avslutar (?) sitt resonemang om kamelerna. Av de stora nyskrivna artiklarna i detta nummer handlar en om adresser i stora datanät. Detta kommer förhoppningsvis att leda till att våra medlemmar får ett större konkret kunskap om världen utanför. En annan artikel handlar om PostScript-koden och hur man kan använda den. Man kan se förhållandet att rena PS-kodfiler numer börjar distribueras som en bakgrund varför denna artikle har skrivits just nu. Vidare har vi tagit med två FAQ-ar som handlar om teckensnitt. Dessa brukar vara mycket djuplodande och vara mycket detaljerade. En förutsättning för att kunna publicera dem är att den mesta texten måste vara med, helst utan ingrepp. Detta gör dem dock mycket svåra att redigera för svenska förhållanden. Layouten på dessa amerikanska texter är då ett sätt att utan att göra alltför stora ingrepp i textmassan framhålla vad som kan anses som väsentligt och ha med vad som skulle kunna kallas "kunskapstext" och ändå ha med detaljerna och bakgrundsmaterialet, om än i mindre teckensnitt.

Ulf Sjöstrand



Mr. Diomidis Spinellis
This is a message from a
n international terrori
st group. We have abd
ucted your files and ha
ve hidden them away. Y
ou must pay $500 by Sun
day afternoon in order t
o see them again. You w
ill receive further telep
hone instructions. Do not no
tify the police or any author
ities; You will endager thei
r life.
Remember: time flies!

Resultatet av programexemplet i artikeln PostScript av Bo Kullmar

Hur skickar jag brev via InterNet och FidoNet från ABC-klubben och hur skickar andra brev till mig?

ABC-Klubbens MSG-system körs på en Unixdator. De flesta datorer anslutna till Internet är Unixdatorer. Brevhanteringen i ABC-Klubbens Unixsystem är baserad på den Unixstandard som används inom Internet. ABC-Klubbens Unixdator är dessutom en Fidonet-nod vilket innebär att datorn har både en Internetadress och en Fidonetadress.

Skall man skriva ett nätbrev på ABC-klubbens system gör man det på Unixsätt eftersom det är ett Unixsystem. Den koppling som har gjorts för brev till yttervärlden har gjorts för Unix-mail. Den programvara som klubben kör för att prata med Fidonet heter rfmail.

Kommandon för brevskrivning

För att skicka ett nätbrev till någon utanför ABC-klubben skall man först och främst ge kommandot

```
BREV 1
```

i MSG. Sen fyller man i adressatens dataadress i Internet-nätet:

```
bk@kullmar.se
```

sen skriver man brevet på vanligt sätt och avslutar med Ctrl-Z. I detta fall kommer brevet att skickas till Bo Kullmars privata dator som han har stående i sängkammaren i sin lägenhet.

Om InterNet-adresser

".se" som står sist i adressen betyder Sverige, stod det ".edu" betyder det ett universitet, vilket som helst, i USA. Lägg märke till att adressen skall skrivas precis som den står här ovan, alltså med små bokstäver, snabel-a, och punkten på rätt ställe. Använder du svensk s k 7-bits teckenuppsättning så blir det É i stället för @. Tidigare användes fysiska adresser på Internet. Man angav maskinnamn åtskilda av utropstecken. Denna typ av adressering bör inte längre användas. Nackdelen med den gamla adressformen var att man var tvingad att hålla reda på den fysiska väg som brevet skulle ta.

Domänadresserna idag är alltså mera en typ av logiska adresser. För att riktigt krångla till det finns det domäner som heter samma som en maskin. "kullmar.se" är ett sådant exempel.

Får man post från kullmar.se händer det att det står

```
"...@kullmar.kullmar.se"
```

som avsändaradress. Det beror på att sendmail på SUN tycker att det första kullmar är maskinnamnet och det andra är domännamnet så att den skapar en adress av det. Man behöver dock inte skriva brev till "bk@kullmar.kullmar.se" utan "bk@ kullmar.se" räcker mycket väl. Skall man skriva brev till Bo Kullmar från MSG på detta gamla sätt så skulle man skriva brev till "monitor!kullmar!bk". bk är Bosses användarid. Monitor är namnet på ABC-klubbens maskin och kullmar är namnet på Bosses dator.

Egentligen hanteras posten på ABC-klubbens system av en fiktiv maskin vid namn "msg". Post till ABC-klubben från Bosses unixdator med fysisk adress, som inte ska användas numera, skulle alltså gå till "monitor!msg!1789", där numret på slutet är Bosses medlemsnummer i ABC-klubben.

Kan man skicka brev till en icke-InterNet-adress via InterNet?

Nu kanske det kan vara litet onödigt att skriva brev till Bo Kullmar via Internet eftersom han ju redan har en brevlåda i ABC-klubben. Betydligt roligare är kanske att skriva brev via Compuserve till USA:s president Bill Clinton. Samma sak gäller, det går alltså till likadant att skicka brev till Bo Kullmar som till president Bill Clinton:

```
brev 1
```

vilket gör att du kommer till brevskrivningsläget, sedan Bill Clintons dataadress i Compuserve-nätet:

```
75300.3115@compuserve.com
```

och så avslutar vi som vi alltid gör när vi har skrivit klart med Ctrl-Z. Compuserve-adressen "75300,3115" skrivs som "75300.3115@compuserve.com" när man skriver från ett Internetsystem som här i ABC-klubben.

Observera att det är ett kommatecken i compuserve-adressen och att detta i Internetformen har bytts ut till en punkt. Men detta är ju inte Internet! Bill Clinton har en dataadress i nätet Compuserve, som är ett kommersiellt nät. Hur kommer det sig då att vi kan skicka brev till Compuserve? Mycket enkelt, häng med.

Adresser i Internetvärlden består alltid av<användare>@domän.domän.domän. topdomän.

Vad är en "domän" i samband med dataadresser?

En domän är ett namn på en organisation eller företag. En topdomän kan även vara ett land, som Sverige. Enligt svenska akademins ordlista är domän liktydigt med område. Det är kanske en bättre förklaring till betydelsen av begrepet domän i sammanhanget.

Det kan finnas en eller flera domäner. Topdomän kan t.ex. vara ".com", eller ".edu", eller ".se", eller ".fi" eller ".us" eller någon annan topdomän. ".com" används av kommersiella företag i USA, ".edu" används av universitet i USA. De andra exemplen pekar på länder. Domänen ".us" används av småföretag i USA.

Till andra datanät

Adresser som används inom Internet men som skall vidare till andra system uttrycks på ett sätt som fyller kravet för Internet, och kan omvandlas till det mottagande systemets egen syntax. Därför har adresser till Compuserve och MSG de medlemsnummer som används i dessa system som användarid. Resterande delen av adresserna är helt vanliga Internetadresser: "compuserve.com" och "msg.abc.se."

Exempelvis så ser min Internetadress ut på följande vis:

```
7872@msg.abc.se
```

Där den första siffergruppen är mitt medlemsnummer i ABC-klubben, eftersnabel-a kommer "msg", som är en domän, ".abc" är också en domän, samt ".se" är topdomänen. Om adressen läses bakifrån blir det så här: först Sverige, sen abc, sen msg, och slutligen mitt medlemsnummer. På så sätt leder adressen fram till just mig. På inloggningsbilden när man ringer till ABC-klubben står det exempel på datadresser. Internetexemplet ser på denna bild ut så här:

```
"abc.se (<mnr>@msg.abc.se"
```

Detta skall tolkas som att domänen för ABC-klubben är "abc.se" och att adressen för medlemmar i ABC-klubbens MSG-system är "`7872@msg.abc.se`". Siffrorna i exemplet är mitt medlemsnummer. Varje medlem i ABC-klubben har sitt medlemsnummer och det är detta medlemsnummer som skall stå först i adressen.

Om FidoNet-adresser

Nu kan ju vi ABC:are nås, och skicka, brev även via Fidonet. En Fidonetadress ser helt annorlunda ut, min adress ser ut så här:

```
2:201/235
```

och det gör samtligas ABC:ares Fidonetadresser. Så hur går det egentligen till att få breven till rätt person? Mycket enkelt, en Fidonetadress är helt enkelt en dataadress till en speciell dator. En s k nod, eller point. För att brevet skall komma rätt person tillhanda så kräver en Fidonetadress att man även skriver namnet på mottagaren. Fidonetadressen skall alltså kombineras med namnet på adressaten, som det är registrerat i databasen på fidonet-datorn dit det ska.

Min Fidonet-adress här i ABC-klubben är alltså:

```
    2:201/235
    Lars Strömberg
```

Följande adress är felaktig:

```
    2:201/235
    Lars L. Strömberg
```

Även denna adress är felaktig:

```
    2:201/235
    Lasse Strömberg
```

Mitt eller någon annans namn måste skrivas precis som det är registrerat i ABC-klubben. Detta gäller även andra datorer som är anslutna till fidonet.

Men hur går det då att skicka brev via Fidonet till utlandet om man som jag har den för Sverige unika bokstaven Ö i namnet? ABC-klubbens MSG använder sig av teckenuppsättningen ISO 646 SE eller vad som annars kallas svensk 7-bits ASCII. Detta innebär att MSG anger Lasses efternamn som Str|mberg. "|" innebär ett problem i Unix eftersom det används som tecken för rör eller pipe i unixskalen. Detta gör att det förmodligen inte fungerar att skicka brev från Fidonet till Lasse om vi inte manuellt ändrar Lasses efternamn i Unix till Stromberg.

Lasse har mottagit brev från en Fidonetnod i frankrike, vid detta tillfälle hade avsändaren i Frankrike skrivit Stromberg i stället för Strömberg, och då funkade allt till full belåtenhet.

Nod och point?

Vad är en nod och en point? Jo, en nod är en dator som ringer till andra noder, medan en point är en dator som ringer till en speciell nod, och bara till denna specella Nod. Här hos ABC-klubben har vi ingen Point, utan vår dator är en Nod i detta sammanhang.

I MSG skriver man fidonet-adressen

```
"Martin Lundberg@f124.n201.z2
.fidonet"
```

för att skriva brev via fidonet till Martin Lundberg på 2:201/124. Skulle Martin ha haft en point, t.ex. nr 2, så skulle man skriva tilll

```
"Martin   Lundberg@p2.f124.
n201.z2.fidonet".
```

Vill man skriva till Martins BBS via ABC-klubben utifrån Internet skriver man till

```
"Martin Lundberg@f124.n201.z2
.fidonet.abc.se"
```

Varför så krångligt när man skicka ett Fidonetbrev från ABC-klubben? Det är ju inte alls så krångligt att skicka ett Fidonetbrev från en Fidonet-BBS till en annan Fidonet-BBS. Svaret är enkelt, det beror på att vi använder en Unixdator.

Vad betyder p, f, n, och z i exemplet på FidoNet-adressen härovan?

"p" betyder point, "f" betyder fidonet, "n" betyder node och "z" syftar på region. Här i ABC-klubben är det noga med dessa bokstäver. Se till att de hammnar på rätt plats i adressen när du skriver till någon i Fidonet. Missar du en enda av dessa bokstäver så kommer vår dator att skicka brevet till Swipnet AB.

Det finns listor med system till vilka man kan skicka brev. Vad gäller brev från ett Internetsystem till ett annat system så går det alltså till så att man på något sätt anpassar adressen till den adress som används i Internet. Exemplen från Compuserve och Fidonet ovan är exempel på detta.

Koppling mellan olika datanät

En liten förklaring om det här med kopplingar mellan olika datanät och de till utseendet olika adresserna

Man får skilja på fysiskt ihopkopplade nät och nät som har samma form av system och addresstyp. Egentligen är Internet ett globalt nät som består av ihopkopplade lokala nät. Gemensamt är att man pratar på nätet med TCP/IP protokoll. ABC-Klubben har ingen direkt förbindelse med detta nätverk eftersom det är för dyrt. Däremot har vi kontakt med Internet via uppringbar förbindlse vilket gör att vi kan ta emot och skicka post som alla andra på "riktiga" Internet.

Övriga nättjänster på Internet fungerar alltså inte mot ABC-Klubben. Man kan alltså inte från Chalmers i Göteborg via nätverket Internet logga direkt in på ABC-Klubben. Man får ta och ringa via modem i stället.

Däremot går det utmärkt att skicka post, vår post hanteras av SWIPNET AB och på vägen till ABC-Klubben tar den vägen över Bo Kullmars Unixdator. Man betraktar system som kan prata med varandra på samma sätt, som ett nät. Fidonet är ett exempel på detta. Ingen Fidonetbas är nämligen ihopkopplade med ett "riktigt" nät utan alla fidonetdatorer ringer till varandra. Man kallar det ändå nät eftersom post och filer kan hanteras med samma adressteknik.

Kan jag från en Fidobas i, säg Danmark, skicka ett brev till min interNet-adress här på ABC-klubben?

Ja, i princip kan du skicka brev från hela världen till ABC-Klubben via Fidonet. Den programvara som ABC-Klubben använder gör detta möjligt och skulle göra det möjligt att skicka breven via ABC-klubbens nodenummer i Fidonet ut till Internet. Nu fungerar inte detta på grund av att Rfmail är buggit. Det finns dock ett annat system med rfmail i Göteborg som drivs av Tomas Tengling (som också är medlem), som kan användas som gateway mellan Fidonet och Internet. Man får dock komma ihåg att eftersom brev i Fidonet förmedlas via privata BBS:er och ägarnas telefonräkningar så finns det ingen garanti för att internationell post går fram. Gateway är en koppling mellan olika datanät. Enligt ordboken betyder gateway ingång eller utgång, och det säger väl ganska väl vad det är frågan om.

Kan jag från en dator ansluten till InterNet skicka ett brtev till en mottagaare med en Fido-adress?

Ja, via Tomas Tenglings gateway. Det borde också fungera att skicka från Internet till Fidonet via ABC-Klubben, men inte i andra riktningen.

Hur ser min adress ut för en amerikanare, austaliensare? Ser den likadan ut som för en ABC-medlem i Osby eller Boden?

Svaret är ja. Din Internetadress här i ABC-klubben ser lika dan ut oavsett var i världen man befinner sig. En person i USA eller Australien, eller Osby eller var som helst på vårt kära gamla klot skriver din adress likadant som alla andra medåkare på detta klot. Även Fidonetadressen ser exakt lika ut oavsett varifrån på jorden brevet sänds. Men, se upp, har du ett Å, Ä eller Ö i ditt namn kan det bli trassel. Då är det inte säkert att brevet kommer fram om det har sänds via Fidonet. Även detta går att lösa via ett s.k. 'alias'. Väntar du post från Fidonet, men har Å, Ä eller Ö i ditt namn så prata med någon som kan ABC-klubbens Unix-system.

Om teckenuppsättningar inom InterNet

Internet tillåter inte post med 8-bitars teckenuppsättning. Man skall nöja sig med de vanliga ASCII upp till 127. Vissa system godtar dock att man skicka 8-bitars tecken, men man kan aldrig vara tvärsäker att det fungerar hos mottagaren. Arbete pågår inom Internet för att även 8-bitars teckenuppsättning skall fungera säkert och felfritt. Inom Fidonetvärlden används både 7-bits teckenuppsättning som inom Internet, samt även den vanliga PC 8-bitars teckenuppsättning.

Vett och etikett

Jaha...hur går det? Har ni hängt med i svängarna? Nu är det inte mycket kvar, så nu kommer en liten överkurs om vett och etikett. Det är inte mycket så häng kvar med läsandet en stund till. En mottagare av ett Internetbrev får normalt upp din avsändaradress om han/hon gör 'kommentera' i sitt brevsystem (ex.v. Lmsg). Det är ändå praxis att längst ner i sitt brev på Internet ange sin avsändaradress och ibland sin postadress och telefon.

En signatur kan se ut så här:

ringer upp till ABC-klubben med LMSG så kommer brevet att skickas iväg ut på Internet eller Fidonet.

Kan LMSG ta emot ett inkommet nätbrev?

Om jag får ett nätbrev sänt till min brevlåda i ABC-klubben, kan jag då använda Lmsg för att läsa brevet? Visst, Lmsg gör ingen skillnad på vanliga brev eller nätbrev. Jag har testat att ge kommandot 'kommentera' när jag läst ett nätbrev i Lmsg, och det har gått utmärkt när det gällt post från Internet, kommentaren har kommit avsändaren tillhanda. Fidonet har jag inte testat detta kommando på.

Har det klarnat?

Det här med dataadresser är verkligen inte lätt, det finns mängder med förvirrande uttryck och begrepp, var därför inte rädd att ställa frågor i MSG. Inga frågor är för 'dumma', och tänk på att det finns alltid någon annan medlem som vill ställa samma fråga, men inte törs eller kan. Så tveka inte,

```
Bo Kullmar, Helsningoersg. 38, 164 44 KISTA, Sweden, Phone +46 8 7511518
Internet: 1789@msg.abc.se                                Fidonet: 2:201/235.0
```

Observera att ö inte har användts i gatuadressen, skickas brevet inom Sverige så kan naturligtvis de nationella bokstäverna användas.

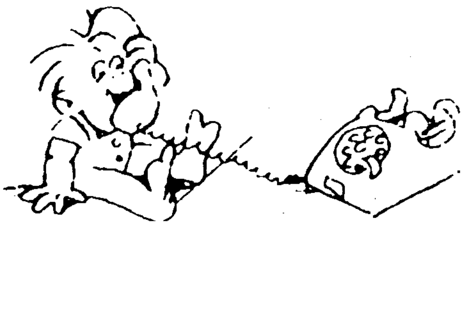
Hur gör vi LMSG-användare när vi skickar nätbrev?

Vi som använder det förnämliga, och för oss ABC:are exklusiva programmet LMSG gjort av Anders Franzén, kan vi använda Lmsg för att skriva brev till mottagare med dataadresser i Internet? Svaret är ja! Det är till och med väldigt enkelt.

Tryck på tangenten B och som resultat därav kommer det upp en liten ruta på skärmen där du ställer markören på alternativet "Brev till person i nät", tryck nu <Ret> och ytterligare en ruta kommer upp. I denna ruta ska du fylla i namnet på mottagaren, <Ret> igen får till resultat ytterligare en ruta där du nu ska fylla i mottagarens dataadress, gör det och tryck på <Ret> igen.

Saken är klar. Har du nu fyllt i namn och dataadress enligt konstens alla regler så är det bara att skriva ditt brev som vilket inlägg som helst i Lmsg, avsluta brevet med det gamla vanliga CTRL-Z. Nästa gång du fråga på, du är inte ensam att tycka att dataadresser och deras användning är förvirrande. För att inte tala om denne skribent, maken till förvirrande ämne att skriva förståerligt om finns nog inte. Jag är långt ifrån säker på att ha lyckats förklara så alla förstår. Skäll i så fall inte på mig, låt oss gemensamt försöka reda ut begreppen i MSG, eller skriv frågor till ABC-bladet.

<7872>Lars Strömberg
med ovärderlig hjälp från
<1789> Bo Kullmar



Trollkarlens lärling

Jag hörde någon säga häromdagen att svenskarna är ett färggrant folk:

De röstar rött, t
änker blått,
äter grönt och
arbetar svart.

Det där sista är kanske inte särskilt underligt. Jag har ibland mediterat över hur mycket muskelkraft som för inte så länge sedan gick åt för att göra alla möjliga saker som vi numera får uträttade "utan att det märks": hämta vatten, söka upp, samla in, hugga och bära ved, elda i diverse ugnar och kaminer, bära bort och tömma latrintunnor osv. I dag har vi låtit maskiner och tekniska underverk ta hand om det mesta av detta. I gengäld har vi satt ett så högt pris på den mänskliga arbetskraften att det inte längre är ekonomiskt möjligt att betala en hantverkare för att fixa litet småsaker omkringen. Det måste man göra själv, om man inte kan ordna det svart (eller möjligtvis helvitt, dvs ideellt).

Okänd kompositör

För ungefär hundra år sedan skrev den franske kompositören Paul Dukas ett scherzo som fick namnet La apprenti sorcier (Trollkarlens lärling). Dukas hade oturen att leva i skuggan av mera namnkunniga samtida och detta är i stort sett hans enda spelade verk. Ironiskt nog har det blivit bäst känt som musiken till Walt Disneys mera kända tecknade långfilm Fantasia, med Musse Pigg som lärlingen. Även "förlagan" och inspirationsämnet är mera känt, dikten Der Zauberlehrling av Goethe, som i sin tur byggde på en gammal saga.

En trollkarl skulle gå ett ärende och sa åt sin lärling att hämta vatten under tiden. Detta var givetvis före vattenledningarnas och elektricitetens tid, så det skulle medföra långt och hårt arbete, vilket väl också var meningen kan man tänka.

Lärlingen var litet lat, som de flesta av oss, och det föll honom in att det nog borde finnas en trollformel i mästarens böcker någonstans, som skulle kunna underlätta det besvärliga jobbet. Han hittade mycket riktigt en formel med vilken han kunde befalla kvastskaftet att hämta vatten i hans ställe.

Formeln fungerade, han fick fart på kvastskaftet och kunde i godan ro ägna sig åt trevligare sysselsättningar medan kvasten utförde det tråkiga slitgörat. Förmodligen kände han sig ganska nöjd ett litet ögonblick, som vi alla när vi har satt vår hand på en ny teknik och tycker att vi börjar behärska den. Men efter ett tag börjar han bli orolig. Fatet som skulle fyllas med vatten är fullt, men den dumma kvasten fortsätter oförtrutet att hämta mer. Lärlingen vet inte hur han skall få stopp på trolleriet, och blir allt mer desperat. (Känns situationen igen?)

I sin förtvivlan tar han till yxan och hugger itu kvasten. Men det ger bara ett kort andrum. Sedan reser sig kvasthalvorna och börjar båda bära vatten, så att den sista villan blir värre än den första, som det står i Skriften. Mja, även den situationen är inte helt obekant för den som sysslar med datorer...

Till slut kommer trollkarlen själv hem. Med barsk stämma skickar hade kvastbitarna tillbaka till hörnet där de hör hemma. Vad som hände med lärlingen förmäler varken dikten eller tonsättningen. Antingen fick han sparken, eller också uppskattade hans läromästare lärlingens kreativitet och upptäckarlusta och kanske lärde honom litet mer om trollerikonsternas hemligheter och fallgropar.

Ny dator

Jag har nyligen blivit begåvad med en ny dator. Det är nog inte den allra häftigaste på marknaden, men den häftigaste som en mera beskedlig verksamhet kan anses ha råd att hålla sin förste datorexpert i det lilla formatet med. Alltnog, det lär vara en äkta 486, fast litet strypt här och där för att få ned priset. Och så 6M i RAM-minnet och en jättebamsig hårddisk på hela 100 M och litet till! I varje fall tyckte jag -- som hade 40M förut -- att det var i mäktigaste laget och undrade hur länge det skulle dröja innan den enorma volymen hade fyllts.

För säkerhets skull bestämde jag mig från början för att göra en ordentlig vindsröjning vid övergången från den gamla datorn till den nya. Inga filer skulle flyttas över som jag inte säkert behövde...

Nåja, det är bra med goda föresatser. Så länge den andra burken, en 286 som var häftig en gång i världen (två år är lång tid i datorsammanhang), kan stå kvar inom räckhåll, kan man ju vänta med den slutliga bedömningen av vad man kan ha och vad man kan mista.

Över huvud taget verkar det vara en policy på allt fler ställen att man behåller åtminstone ett ex av en äldre maskin med tillhörande äldre diskar och programmängder när man uppdaterar systemet på platsen.

När jag hade flyttat över det som jag bedömde som aktuellt, fyllde det bara ca 30M. Med 75% kvar på skivan kände jag mig rik och förhoppningsfull.

Visserligen har jag nu och då nödgats hämta ytterligare en eller annan fil, men det har inte stört bilden mycket. Men det var innan jag gav mig i kast med Windows!

WINDOWS?

Alla omkring mig har tjatat länge om att "vi måste ju ha Windows". Jag haft goda skäl att hålla emot. För det första har datorerna i min verksamhet mest använts av handikappade elever som inte kan hantera vare sig mus eller styrkula och som därför inte självklart har någon glädje av ett grafiskt gränssnitt. För det andra har vi inte haft någon skrivare som kunnat skriva ut grafiska sidor. Och för det tredje (och avgörande) har vi i alla fall inte kunnat köra W på våra gamla 88-or.

Dock rullar utvecklingen obönhörligt på. Våra gamla trotjänare blir allt gistnare och det börjar bli läge att byta ut dem. Deras efterträdare blir 386 eller 486 och då finns kanske anledning att övergå till färgskärm, VGA och laserskrivare (om pengarna räcker).

Alltnog, jag som dataansvarig måste gå före och lära mig det nya som ev kommer. Sålunda skaffade jag Windows. Och plötsligt hade jag inga 75% lediga på disken längre, utan bara 40%. Sedan kom PC-TOOLS för W därmed var det 50%.

Parkinsons lag

Parkinsons första lag lyder: utgifterna stiger med inkomsterna. Den gäller tydligen också datorutrymme. Det verkar som om en ny och större hårddisk alldeles av sig själv drar till sig nya och större program. Innan man vet ordet av är det fullt igen! Nya och smarta lösningar med kompressionsprogram som klämmer ihop innehållet till hälften är bara en temporär lösning. Den förhindrar inte, bara uppskjuter några månader det oundvikliga bytet till en ännu större hårddisk. Den i sin tur drar till sig ännu fler bamseprogram osv. Hjälp! Hur skall vi få slut på detta?

Herre eller slav?

Ända sedan industrialismens framväxt har det av och till blossat upp diskussioner om maskinerna som människans herre eller slav. Vi vill gärna tro att det är människan som bestämmer över maskinen. Och naturligtvis är det så, praktiskt sett. Men att maskinen finns där påverkar människans kultur. Utan elektriciteten skulle vi fortfarande hämta vårt vatten i hinkar vid brunnen. Utan bilar skulle vi fortfarande inte ha några ordentliga vägar. Utan datorn skulle vi fortfarande skriva och räkna för hand (om vi nu hoppar över skrivmaskinen och räknesnurran). Vi blir beroende. Visst händer det att man känner sig ganska närbesläktad med trollkarlens lärling. Vi har släppt lösa krafter som växer oss över huvudet och börjar leva sitt eget liv med oss.

Varför blir det så? Varför säger vi inte ifrån hur vi vill ha det? Jag tror svaret är mycket enkelt: vi gör det! Vi vill nämligen ha våra maskiner för den bekvämlighet och de finesser de ger oss. Vi vill åka bil, få vatten ur kranen, ha elapparater i gång överallt och köra våra datorer till nytta och nöje. Vi vill inte kånka hem vatten och ved och gå till fots.

Tillbaka — vart?

Det finns all tid människor som försöker gå emot strömmen. "Utvecklingen var nog bra, men den höll på för länge. Nu måste vi tillbaka till naturen." Heder åt dem -- de försöker åtminstone följa sin övertygelse! Men det är inte lätt att vara konsekvent. Det är svårt att se alla följder av sitt handlande. Man vill gärna ha kvar alla fördelar som maskinerna ger oss, samtidigt som man blir av med nackdelarna. Det är inte så enkelt. Under tiden får vi försöka sansa oss litet. Behöver man alla program som vi fyller våra hårddiskar med? Kanske det i själva verket är med hårddisken som med källarkontoret eller vindsförrådet: man lägger saker där som man strängt taget inte behöver, men som "kan vara bra att ha". Så småningom blir det fullt -- det blir alltid fullt, förr eller senare.

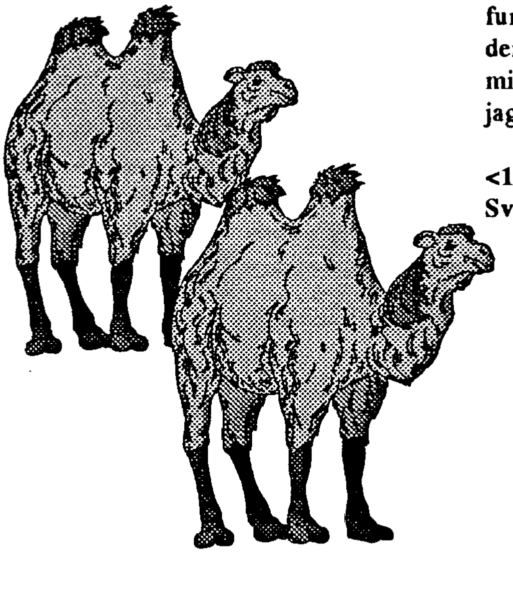
Ingen teknik i världen kan rädda oss från vår egen slöhet. Skall det bli människans mått på en ny teknik är det också människan som måste hålla ut sin måttstock över tekniken och inte tvärtom.

<1384>
Sven Wickberg

Ett förslag till lösning av problemet med kamelerna

Ett förslag till lösning

```
10 ! arab
20 !
90 FLOAT
100 FOR A=2 TO 4
105    Sum=0
110    A(1)=A
120    Sum=Sum+1/A
130    FOR B=3 TO 10
132       IF B<A THEN 270
135       Sumb=Sum
140       Sumb=Sumb+1/B
150       IF Sumb>=1 THEN 270
160       A(2)=B
170       FOR C=3 TO 25
172          IF C<B THEN 260
175          Sumc=Sumb
180          Sumc=Sumc+1/C
190          IF Sumc>=1 THEN 260
200          A(3)=C
210          FOR D=3 TO 50
212             IF D<C THEN 250
215             Sumd=Sumc
220             Sumd=Sumd+1/D
230             A(4)=D
240             ! IF ABS(1-Sumd)<.00001
                   GOSUB 500
241             IF Sumd=1 GOSUB 500
250          NEXT D
260       NEXT C
270    NEXT B
280 NEXT A
281 END
290 !
500 FOR I=1 TO 4
510    ; A(I);
520 NEXT I
530 ;
540 RETURN
```



Kommentarer:

Det här programmet är skrivet helt utan finesser för att det skulle gå snabbt och lätt och inte ge upphov till en massa letande efter buggar.

Jag utgick ifrån att man skulle söka värdena så att a<=b<=c<=d. Då kan a inte vara mycket större än 2, men för säkerhets skull undersöktes även 3 och 4. En enkel analys visar att a under inga förhållanden kan vara 5 eller högre för

$$1/5 + 1/5 + 1/5 + 1/5 < 1$$

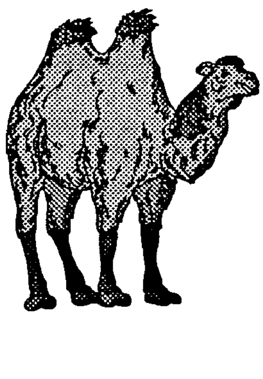
och därmed är alla andra lösningar ännu mindre.

På liknande sätt kan man resonera sig fram till lämpliga maxvärden för b och c. Däremot var det svårare att gissa yttersta för d. Men enligt den gamle bediunhövdingens förutsättningar skulle det röra sig omkring 37. Jag drog till med 50 för säkerhets skull, och det visade sig räcka.

Programmet råräknar sig alltså igenom alla tänkbara kombinationer, och bråkens värde summeras på raderna 120, 140, 180 och 220 successivt. Vi söker värden på a, b, c och d så att slutsumman blir 1.

Rad 240 ger uttryck för mitt tvivel, som sedan visade sig berättigat. Kan man räkna med att alla dessa bråktal verkligen bli EXAKT 1 tillsammans, eller måste man räkna med en slarvmarginal? Raden 240 fungerade inte riktigt bra, så jag övergav den när jag fann att rad 241 prompt försåg mig med hela raden lösningar. (Trodde jag.)

<1384>
Sven Wickberg



Vanliga frågor om teckensnitt

Table of Contents

1. General Information
1.1. Notes about the FAQ
1.2. Font Houses
1.3. What's the difference between type fonts, type 3 fonts,
1.4. What about "Multiple Master" fonts?
1.5. Is there a methodology to describe and classify typefaces?
1.6. What is the "f" shaped "s" called?
1.7. What about "Colonial" Typefaces?
1.8. Where can I get _____ fonts.
1.9. Where can I get fonts for non-Roman alphabets?
1.10. What about fonts with the International Phonetic Alphabet
1.10.1. Shareware or free (PostScript Type 1 and/or TrueType):
1.10.2. Shareware or free (TeX):
1.10.3. Commercial:
1.11. How can I convert my _____ font to _____ format?
1.12. Are fonts copyrightable?
1.13. Typeface Protection
1.14. File Formats
1.14.1. File Format Extensions
1.14.2. Font Formats
1.14.3. Font Format Extensions
1.15. Ligatures
1.16. Standard Laser Printer Fonts
1.17. Glossary
1.18. Bibliography
1.19. (En)Coding Standards
1.20. TrueType
1.21. Rules of Thumb
1.23. A Brief Introduction to Typography
1.23.1. Comments by Laurence Penney:
1.23.2. Comments by Don Hosek:
1.24. What does 'lorem ipsum dolor' mean?

General Information

1.1. Notes about the FAQ

Many FAQs, including this one, are available on the anonymous ftp archive site rtfm. mit.edu in the directory pub/usenet/news. answers.

The name under which a FAQ is archived appears in the Archive-name header at the top of each article.

This FAQ is a work in progress. If you have any suggestions, I would be delighted to hear them. Although I had intended to complete a major reorganization before this posting, it did not come to pass. I have, however, changed the way the FAQ is built (again). This FAQ was constructed by LameTeX.

The FAQ is formatted for easy searching. Most news readers can skip from one question to the next by pressing control-G. (or by searching for "Subject:" in column 1)

The Gopher server on port 70 at ibis.cs.umass.edu provides a WAIS-indexed version of this FAQ.

All trademarks used in this document are the trademarks of their respective owners. Standard disclaimers apply.
The FAQ is maintained by Norm Walsh <walsh@cs.umass.edu>

1.2. Font Houses

This section will be expanded on in the future. It contains notes about various commercial font houses.

1.2.1 Compugraphic
See "Miles, Agfa Division"

1.2.2 Miles, Agfa Division
Compugraphic which was for a while the Compugraphic division of Agfa, is now calling itself "Miles, Agfa Division" (yes, the Miles drugcompany), since CG's off-shore parent Agfa has been absorbed by Miles. So typographically speaking, Compugraphic, CG, Agfa, A-G ag, and Miles all refer to the same company and font library. Their proprietary fonts are still CG Xyz, but the name is Miles Agfa.

1.3. What's the difference between type 1 fonts, type 3 fonts, type 5 fonts, Macintosh fonts, Windows fonts, LaserJet fonts, etc.

This question is not trivial to answer. It's analogous to asking what the difference is between various graphics image file formats.

The short, somewhat pragmatic answer, is simply that they are different ways of representing the same "information" and some of them will work with your software/printer and others won't.

At one level, there are two major sorts of fonts: bitmapped and outline (scalable). Bitmapped fonts are falling out of fashion as various outline technologies grow in popularity and support.

Bitmapped fonts represent each character as a rectangular grid of pixels. The bitmap for each character indicates precisely what pixels should be on and off. Printing a bitmapped character is simply a matter of blasting the right bits out to the printer.

There are a number of disadvantages to this approach. The bitmap represents a particular instance of the character at a particular size and resolution. It is very difficult to change the size, shape, or resolution of a bitmapped character without significant loss of quality in the image. On the other hand, it's easy to do things like shading and filling with bitmapped characters.

Outline fonts represent each character mathematically as a series of lines, curves, and 'hints'. When a character from an outline font is to be printed, it must be 'rasterized' into a bitmap ``on the fly''. PostScript printers, for example, do this in the print engine. If the "engine" in the output device cannot do the rasterizing, some front end has to do it first. Many of the disadvantages that are inherent in the bitmapped format are not present in outline fonts at all. Because an outline font is represented mathematically, it can be drawn at any reasonable size.

At small sizes, the font renderer is guided by the 'hints' in the font; at very small sizes, particularly on low-resolution output devices such as screens, automatically scaled fonts become unredable, and hand-tuned bitmaps are a better choice (if they are available).

Additionally, because it is rasterized "on demand," the font can be adjusted for different resolutions and 'aspect ratios'.

LaserJet .SFP and .SFL files, TeX PK, PXL, and GF files, Macintosh Screen Fonts, and GEM .GFX files are all examples of bitmapped font formats.

PostScript Type 1, Type 3, and Type 5 fonts, Nimbus Q fonts, TrueType fonts, Sun F3, MetaFont .mf files, and LaserJet .SFS files are all examples of outline font formats.

Neither of these lists is even close to being exhaustive.

To complicate the issue further, identical formats on different platforms are not necessarily the same. For example Type 1 fonts on the Macintosh are not directly usable under MS-DOS or Unix, and vice-versa.

It has been pointed out that the following description shows signs of its age (for example, the eexec encryption has been thoroughly hacked). I don't dispute the observation and I encourage anyone with the knowledge and time to submit a more up to date description.

It has further been suggested that this commentary is biased toward Kingsley/ATF. The omission of details about Bitstream (and possibly Bauer) may be considered serious since their software lies inside many 3rd-party PostScript interpreters.

The moderators of this FAQ would gladly accept other descriptions/explanations/viewpoints on the issues discussed in this (and every other) section.

[Ed Note: Liam R. E. Quim supplied many changes to the following section in an attempt to bring it up to date. Hopefully it is a better reflection of the state of the world today (12/07/92) than it was in earlier FAQs]

Henry Schneiker <reachable electronically?> wrote the following description of the differences between several scalable font technologies: There has been a lot of confusion about font technologies in recent times, especially when it comes to Type 1 versus Type 3 fonts, "hints," PostScript compatibility, encryption, character regularizing, kerning, and the like.

*** Encryption (eexec)**

All fonts produced with Adobe's font technology are protected through data encryption. The decryption is provided by the 'eexec' (encrypted execute) PostScript operator and, until recently, was only present in Adobe's licensed PostScript.

Adobe has published the details of the Type 1 font format in the 'Black Book', Adobe Type 1 Font Format (version 1.1), Adobe Systems Inc., 1990. The encryption was mainly used because of font copyright problems; unencrypted fonts can also be used, but these tend to use an efficient binary encoding, also in documented the Type 1 book, and so are still not readable PostScript.

*** Type 1, Type 3, and Type 5 font formats**

There are generally three font formats used in Adobe PostScript printers: Type 1, Type 3, and Type 5. Type 1 fonts are Adobe's downloadable format. Type 3 fonts are third-party downloadable format. Type 5 fonts are the ROM-based fonts that are part of your printer.

There is no functional difference between a Type 1, Type 3, or Type 5 font. A Type 3 font can do anything a Type 1 or Type 5 font can do. The only real difference between them is where the 'BuildChar' routine comes from. For Type 1 and Type 5 fonts it's built into the printer. For Type 3 fonts it's built into the font. In other words, anything a Type 1 font can do a Type 3 font can also do.

[Ed note: the reverse is not true. Type3 fonts can do things that Type1 fonts cannot. But they aren't hinted...]

When PostScript is asked to generate a character, PostScript looks in the font's dictionary for FontType. If FontType is 1 or 5 PostScript executes an internal routine that knows how to interpret the font data stored in CharStrings. If FontType is 3 PostScript executes the routine BuildChar from the font's dictionary to interpret the font data (often stored in CharStrings).

However, each BuildChar routine is written to read data formatted in a method convenient to the vendor. Adobe, Altsys, Bitstream, and Kingsley/ATF all format their font data differently and, hence, have different BuildChar routines.

[Ed note: relative hard disk efficiency of Kingsley vs. Adobe fonts deleted on 12/07/92]

Type 5 fonts are special in that they often include hand-tuned bitmaps for the commonly used sizes, such as 10- and 12-point.

Other sizes are generated from the outlines in normal fashion.

Don't confuse Type 1, Type 3, and Type 5 fonts with Bitstream's Type A, Type B, Type C, and Type F. They are not the same and serve only to confuse the issue.

*** Resolution 'hints'**

When a character is described in outline format the outline has unlimited resolution. If you make it ten times as big, it is just as accurate as if it were ten times as small.

However, to be of use, we must transfer the character outline to a sheet of paper through a device called a raster image processor (RIP). The RIP builds the image of the character out of lots of little squares called picture elements (pixels).

The problem is, a pixel has physical size and can be printed only as either black or white. Look at a sheet of graph paper. Rows and columns of little squares (think: pixels). Draw a large 'O' in the middle of the graph paper. Darken in all the squares touched by the O. Do the darkened squares form a letter that looks like the O you drew? This is the problem with low resolution (300 dpi). Which pixels do you turn on and which do you leave off to most accurately reproduce the character?

All methods of hinting strive to fit (map) the outline of a character onto the pixel grid and produce the most pleasing/ recognizable character no matter how coarse the grid is.

[Ed note: deleted some paragraphs that are no longer true. Times change...]

*** Optical Scaling**

Optical Scaling modifies the relative shape of a character to compensate for the visual effects of changing a character's size. As a character gets smaller, the relative thickness of strokes, the size of serifs, the width of the character, the intercharacter spacing, and interline spacing should increase. Conversely, as a character gets larger, the relative thickness, widths, and spacing should decrease.

Contrast this with linear scaling, in which all parts of a character get larger or smaller at the same rate, making large characters look wide and heavy (strokes are too thick, serifs are too big) while small characters look thin and weak.

*** Kerning**

As applied to PostScript fonts, kerning refers to kern pairs. A kern pair specifies two characters (e.g., A and V) and the distance to move the second character relative to the first. The typical use of a kern pair is to remove excessive space between a pair of characters. However, it may also be used to add space.

* **PostScript clones**

There are currently several printer manufacturers on the market with PostScript clones. To be viable, a PostScript clone must comply with the 'red book' (PS Language Reference Manual). In order to avoid paying royalties to Adobe, and because Adobe's Type 1 font format was originally preprietary, many PostScript interpreters use some other font format. Sun uses F3, and some other vendors use Bitstream's Speedo format, for example. The only real problem this causes is that the widths of characters (the 'font matrics') may vary from Adobe's, so that programs that assume the Adobe character widths will produce poor quality output. Bitstream fonts used to be particularly bad in the early days, but they and most or all of the other vendors have solved those problems.

* **Apple TrueType** ***[Ed note: formerly "Royal ('sfnt')"]*** **format and System 7-**

Apple's new System 7.0 supports a new format of outline font that will allow high-quality characters of any size to be displayed on the screen. TrueType stores font outlines as B-spline curves along with programmed resolution hints. B-spline curves are faster to compute and easier to manipulate than the Bezier curves used in PostScript.

Adobe is not going to support Apple's new format by converting the Adobe/Linotype library to B-spline format. There are two reasons for this: First, there is no support for font encryption (yes, the hooks are there, but nothing is implemented). Second, Adobe does not want to dilute PostScript and its font library. However, the Macintosh is too big a market to simply turn away from. Therefore, Adobe will provide its Font Manager to display its own fonts on the Mac screen. Apple ships Adobe's ATM for this purpose.

1.4. What about "Multiple Master" fonts?

Multiple Master Fonts are an extension to the Adobe font format providing the ability to interpolate smoothly between several "design axes" from a single font. Design axes can include weight, size, and even some whacko notions like serif to sans serif.

Adobes' first Multiple Master Font was Myriad — a two-axis font with WEIGHT (light to black) on one axis, and WIDTH (condensed to expanded) along the other axis. In the case of Myriad, there are four "polar" designs at the "corners" of the design space. The four designs are light condensed, black condensed, light expanded, and black expanded.

Given polar designs, you can set up a "weight vector" which interpolates to any point within the design space to produce a unique font for a specific purpose. So you can get a "more or less condensed, somewhat black face''.

Multiple Master Fonts can be used on any PostScript printer. Multiple Master Fonts need a new PostScript operator known as makeblendedfont. The current crop of Multiple Master Fonts supply an emulation of this operator so the printer doesn't need this operator.

A short tutorial on Multiple Master Fonts and makeblendedfont appears in PostScript by Example, by Henry McGilton and Mary Campione, published by Addison-Wesley.

1.5. Is there a methodology to describe and classify typefaces?

There is an ISO Standard catagorization, P..S something, but it is mostly ignored by typographers; at least one book by a respected authority, Alexander Lawson, Printing Types: An Introduction, had nothing nice to say about it. He has his own, less rigorous system, which is exposited in "An Introduction" and used without exposition in his later "Anatomy of a Typeface". I have another book, Rookledges International Typefinder, which has a very complete system that uses tell-tales of individual glyphs as well as overall style to index most known faces right in the book. J.Ben Leiberman has another book on type face description.

1.6. What is the "f" shaped "s" called?

Both the "f" with half a crosbar (roman) and the integral sign (italic) are called long-S.

1.7. What about "Colonial" Typefaces?

1.7.1 Why does colonial printing have that "Colonial" feel?

Colonial type was either very roughly treated by moist salt air on the crossing and in colonial port cities, or was copied locally by tacky techniques (such as driving used foundry type into soft lead to make very soft deformable matrices), and the paper was very rough, which abrades both the serifs and the hairlines. So except for the best work done with new, european types, the serifs were much smaller, even broken off, than the original founder/punchcutter intended. Thins could be abraded by rough paper to nothingness, esp after humid salt air had leached the hardner out of the alloy.

1.7.2 What fonts are good for mock-colonial uses?

For example, what fonts have the following features: old-style figures (non-lining numbers), the long s character, slightly irregular shapes (a la type produced by colonial printers), and a decent complement of ligatures. And what about free or cheap faces like this?

I don't know if any exist with all of 1-5. As I believe you get what you pay for, especially in fonts, I haven't looked at free and cheap-copy fonts.

Microsoft's expansion set for their Win3.1 optional fonts has Garamond Expert & Expert Extensions, which has a good complement of ligatures and I think I remember it haveing the long ess too. I forget about OSFigs; it should tho'. Monotype's metal faces ``16th Century Roman'' and ``Poliphilus'' may be available in digital; if so, they imitate early presswork with early and are very close to what one wants.

"A commercial supplier *[not yet sampled]* is Image Club Graphics in Calgary (1-800-661-9410). It is called Caslon Antique. It is supplied as both roman and italic, together, for 25. They advertise in MacWorld/MacUser/MacBlah. I am unable to tell from abcDEF123 if the numerals are old-style, but I think not. Ligatures? long-S? Not yet known. Guillemots, though, are there. ... Letraset, circa 1977, showing a Caslon Antique with modern numerals, no ligatures, and only UKPounds and German ss extensions.''

NB: Caslon Antique is not a Caslon per se: "The last Caslon to mention is that ubiquitous but unrelated Caslon Antique, which possesses no similarity whatsoever to the original. This old reprobate was introduced by Barnhart Brothers of Chicago under the name Fifteenth Century. Its negative reception lasted until about 1918, when, with a simple name change to Caslon Antique, it became the most commonly selected type for reproductions of colonial American printing. It is now seen in everything from liquor advertisments to furniture commercials''

Miles Agfa (Compugraphic) has always had a Caslon Antique; I don't know if it is available for TrueType or Type 1, but Agfa has been doing TrueType bundles at reasonable prices.

1.7.3 What fonts could a colonial printer have had?

According to D.B.Updike in the classic reference "Printing Types: Their History, Forms & Use", he indicates that most colonial work was with types of the Caslon Old Style fonts and cheap copies of same in the 18th C. Before that, it would have been the older Dutch & English faces, almost always lagging English tastes. If you can find the Oxford Fell types, they are classic Dutch-as-used-by-englishmen. Anything with a Dutch moniker and the Oldstyle adjective is probably ok; Van Dijck if you find it, say (died 1673).

Ben Franklin recommended Caslon faces. But these were not available in England before 1720, first full broadside in 1734. Lawson declares that the first printing of the Declaration of Independance was in Caslon.

Wilson's Scotch Modern was the ``modern'' font that surfaced in quantity in america. If the Scotch Roman your vendor has is sort-of like-Bodoni but nicer than his Bodoni, that's it. It wasn't available until late 1700s, though.

1.8. Where can I get __ fonts.

Before I go any farther, let me extol the virtues of the Archie servers. If you need to find something on the net, and you have any idea what it might be called, Archie is the place to go. In North America, telnet to "archie.rutgers.edu" and login as "archie". There are many other servers around the world, any Archie server can give you a list of other servers. There are better documents than this to describe Archie and you should be able to find them from the above starting point. If you have trouble, feel free to ask norm <walsh@cs .umass.edu> (via Email please, no need to clutter comp.fonts with a query about Archie ;-).

In addition to the telnet option, several archie clients exist including a very nice X11 implementation (Xarchie)

- **Adobe Type 1 Fonts in MS-DOS/Unix Format:**
  ftp.cica.indiana.edu:/pub/pc/win3/fonts
  ftp.cica.indiana.edu:/pub/pc/win3/fonts/atm
  archive.umich.edu:/msdos/mswindows/fonts

- **Adobe Type 1 Fonts in Mac Format:**
  mac.archive.umich.edu:/mac/system.extensions/font/type1
  sumex-aim.stanford.edu:/info-mac/font

- **Adobe Type 3 Fonts in Mac Format:**
  mac.archive.umich.edu:/mac/system.extensions/font/type 3

- **TrueType fonts in MS-DOS Format:**
  ftp.cica.indiana.edu:/pub/pc/win3/truetype

- **TrueType fonts in Mac Format:**
  mac.archive.umich.edu:/mac/system.extensions/font/truetype

- **TeX PK/PXL/GF fonts:**
  The TeX community has it's own support groups that can provide better answers to this question. The canonical list of MetaFont fonts is posted occasionally to comp.text.tex. The comp.text.tex newsgroup (or the Info-TeX mailing list, if you do not have access to news) are good places to start. Email norm <walsh@cs.umass.edu> if you need more specific information.

- **LaserJet bitmap fonts:**
  wuarchive.wustl.edu:/mirrors/msdos/laser Also on other simtel20 mirrors...

If you know of other archive sites (the above list is no where near complete) or other formats that are available on the net, please let us know.

The sites above represent places where shareware and public domain fonts are available. Many, many typefaces are not available in shareware form. And many shareware faces are less than adequate for a variety of reasons, particularly at small sizes. It seems to be the consensus of the comp.fonts community that "you get what you pay for." If you need a professional quality font, you should probably buy it from a professional.

A list of font vendors (annotated with information about non-Roman alphabets) was contributed by Masumi Abe <abe@adobe .com>. Masumi is Adobe's Manager of Typographic Marketing for Asia. *[ed: as of 7/92]*

The list is quite long and it is posted separately. It can be retrieved via anonymous ftp from /pub/norm/comp.fonts on ibis.cs.umass.edu.

1.9. Where can I get fonts for non-Roman alphabets?

As mentioned above, the list of font vendors is annotated with information about non-Roman alphabets. Commercially, Masumi <abe@adobe.com> suggests that Linguists' Software is the current *[ed: as of 7/92]* leading supplier of non-Roman fonts.

1.10. What about fonts with the International Phonetic Alphabet (IPA) symbols?

1.10.1. Shareware or free (PostScript Type 1 and/ or TrueType):

- **TechPhon**
  Seems to lack some characters and has no zero-offset characters (for accents).

- **PalPhon**
  A phonetic font which you can get by anonymous ftp from mac.archive.umich .edu. It is called PalPhon. There are actually two fonts: the basic PalPhon and one with additional accents and symbols called PalPi. The package includes some documents on using the fonts as well.

- **SIL-IPA**
  SIL-IPA is a set of scalable IPA fonts containing the full International Phonetic Alphabet with 1990 Kiel revisions. Three typefaces are included:

- **SIL Doulos (similar to Times)**

- **SIL Sophia (similar to Helvetica)**

- **SIL Manuscript (monowidth)**
  Each font contains all the standard IPA discrete characters and non-spacing diacritics as well as some suprasegmental and puncuation marks. Each font comes in both PostScript Type 1 and TrueType formats. The fonts are also available for Microsoft Windows. These fonts were designed by the Printing Arts Department of the Summer Institute of Linguistics, Dallas, Texas.

1.10.2. Shareware or free (TeX):

METAFONTsources of the phonetic symbols developed by Tokyo-Shoseki-Printing and Sanseido are available. The font contains all of IPA (Internatioanl Phonetic Alphabet) symbols.

You can get phonetic symbols METAFONT (named TSIPA) from ftp.foretune.co.jp:/pub/tools/TeX/Fonts The IP address for ftp.foretune.co.jp is 133.123.1.2.

1.10.3. Commercial:

Linguist's Software Adobe (ITC Stone Phonetic [255] , Times Phonetic [278])

1.11. How can I convert my ___ font to ___ format?

Conversion from one bitmapped format to another is not generally too difficult. Conversion from one scalable format to another is very difficult. Several commercial software packages claim to perform these tasks, but none has been favorably reviewed by the comp.fonts community. ATech's AllType program, in particular, has had poor reviews *[ed: as of 7/92]* .

For specific conversions, check the platform specific parts of the FAQ. Most of the conversions discussed require platform specific tools.

Here is a summary of the conversions discussed (and the section in which they appear):

From	To	Notes
Mac Type1 PostScript	PC Type1 PostScript	MS-DOS
PC Type1 PostScript	Mac Type1 PostScript	Mac, commercial
TrueType	Type1 PostScript	> No answer as
Type1 PostScript	TrueType	> of 7/92
PC Type1 PostScript	TeX PK	MS-DOS
TeX PK	HP LaserJet bitmaps	MS-DOS
HP LaserJet bitmaps	TeX PK	MS-DOS
TrueType	HP LaserJet bitmaps	MS-DOS, hack!!

In addition, Adobe ships a copy of Adobe Font Foundry with all of its fonts which can convert Type 1 fonts into HP LaserJet softfonts.

1.12. Are fonts copyrightable?

This topic is hotly debated at regular intervals on comp.fonts. Terry Carroll <tjc50@juts.ccc.amdahl.COM> provides the following analysis of current [ed: as of 6/92] legislation and regulation regarding fonts and copyrights. Members of the comp.fonts community are encouraged to submit other materials that add clarity to the issue.

It has been pointed out that this section deals primarily font copyright issues relevant to the United States and that this situation is not universal. For example, in many parts of Europe typeface designs are protectable.

"First, the short answer in the USA: Typefaces are not copyrightable; bitmapped fonts are not copyrightable, but scalable fonts are copyrightable. Authorities for these conclusions follow.

Before we get started, let's get some terminology down:

A typeface is a set of letters, numbers, or other symbolic characters, whose forms are related by repeating design elements consistently applied in a notational system and are intended to be embodied in articles whose intrinsic utilitarian function is for use in composing text or other cognizable combinations of characters.

A font is the computer file or program that is used to represent or create the typeface.

Now, on to the legal authorities:

Volume 37 of the Code of Federal Regulations specifies this about the copyrightability of typefaces:

"The following are examples of works not subject to copyright and applications for registration of such works cannot be entertained: . . . typeface as typeface" 37 CFR 202.1(e).

By the way, you won't find that in the most recent (7/1/91) edition of the CFR; the addition was enacted 2/21/92. It'll be in the next edition, though. It's described in the 2/21/92 edition of the Federal Register, page 6201 (57 FR 6201). The change didn't actually change the law, it just clarified it, and codified existing Copyright Office policy.

The regulation is in accordance with the House of Representatives report that accompanied the new copyright law, when it was passed in 1976:

"The Committee has considered, but chosen to defer, the possibility of protecting the design of typefaces. A 'typeface' can be defined as a set of letters, numbers, or other symbolic characters, whose forms are related by repeating design elements consistently applied in a notational system and are intended to be embodied in articles whose intrinsic utilitarian function is for use in composing text or other cognizable combinations of characters. The Committee does not regard the design of typeface, as thus defined, to be a copyrightable 'pictorial, graphic, or sculptural work' within the meaning of this bill and the application of the dividing line in section 101." H. R. Rep. No. 94-1476, 94th Congress, 2d Session at 55 (1976), reprinted in 1978 U.S. Cong. and Admin. News 5659, 5668.

It's also in accordance with the one court case I know of that has considered the matter: Eltra Corp. V. Ringer, 579 F.2d 294, 208 USPQ 1 (1978, C.A. 4, Va.).

The U.S. Copyright Office holds that a bitmapped font is nothing more than a computerized representation of a typeface, and as such is not copyrightable:

"The *[September 29, 1988]* Policy Decision *[published at 53 FR 38110]* based on the *[October 10,]* 1986 Notice of Inquiry [published at 51 FR 36410] reiterated a number of previous registration decisions made by the *[Copyright]* Office. First, under existing law, typeface as such is not registerable. The Policy Decision then went on to state the Office's position that 'data that merely represents an electronic depiction of a particular typeface or individual letterform' *[that is, a bitmapped font]* is also not registerable." 57 FR 6201.

However, scalable fonts are, in the opinion of the Copyright Office, computer programs, and as such are copyrightable:

"...the Copyright Office is persuaded that creating scalable typefonts using already-digitized typeface represents a significant change in the industry since our previous *[September 29, 1988]* Policy Decision. We are also persuaded that computer programs designed for generating typeface in conjunction with low resolution and other printing devices may involve original computer instructions entitled protection under the Copyright Act. For example, the creation of scalable font output programs to produce harmonious fonts consisting of hundreds of characters typically involves many decisions in drafting the instructions that drive the printer. The expression of these decisions is neither limited by the unprotectable shape of the letters nor functionally mandated. This expression, assuming it meets the usual standard of authorship, is thus registerable as a computer program." 57 FR 6202."

1.13. Typeface Protection

[*Ed: The following article was written by Charles Bigelow several years ago. In the past, I have resisted including this in the FAQ because I do not know the status of any copyrights that may exist on it. However, I've decided to assume that Mr. Bigelow would like to see it distributed as widely as possible so it's more-or-less acceptable to include it here. If anyone knows of a reason why I shouldn't include it, please let me know. norm 2/93*]

1.13.1 Preamble

The main question of typeface protection is: Is there anything there worth protecting? To that the answer must certainly be: Yes. Typeface designs are a form of artistic and intellectual property. To understand this better, it is helpful to look at who designs type, and what the task requires.

1.13.2 Who Makes Type Designs?

Like other artistic forms, type is created by skilled artisans. They may be called type designers, lettering artists, punch-cutters, calligraphers, or related terms, depending on the milieu in which the designer works and the technology used for making the designs or for producing the type.

Type designer and lettering artist are self-explanatory terms. Punch-cutter refers to the traditional craft of cutting the master image of a typographic letter at the actual size on a blank of steel that is then used to make the matrix from which metal type is cast. Punch-cutting is an obsolete though not quite extinct craft. Seeking a link to the tradition, modern makers of digital type sometimes use the anachronistic term digital punch- cutter. Calligrapher means literally one who makes beautiful marks. The particular marks are usually hand-written letters, though calligraphers may design type, and type designers may do calligraphy.

It usually takes about seven years of study and practice to become a competent type designer. This seems to be true whether one has a Phd. in computer science, an art-school diploma, or no academic degree. The skill is acquired through study of the visual forms and practice in making them. As with geometry, there is no royal road. The designing of a typeface can require several months to several years. A family of typefaces of four different styles, say roman, italic, bold roman, and bold italic, is a major investment of time and effort. Most type designers work as individuals. A few work in partnership (Times Roman(R), Helvetica(R), and Lucida(R) were all, in different ways, the result of design collaboration.) In Japan, the large character sets required for a typeface containing Kanji, Katakana, and Hirakana induce designers to work in teams of several people. typeface family is an accomplishment on the order of a novel, a feature film screenplay, a computer language design and implementation, a major musical composition, a monumental sculpture, or other artistic or technical endeavors that consume a year or more of intensive creative effort. These other creative activities can be protected by copyright or other forms of intellectual property protection. It is reasonable to protect typefaces in the same way.

1.13.3 The Problem of Plagiarism

A lack of protection for typeface designs leads to plagiarism, piracy, and related deplorable activities. They are deplorable because they harm a broad range of people beyond the original designers of the type. First, most type plagiarisms are badly done.

The plagiarists do not understand the nature of the designs they are imitating, are unwiling to spend the necessary time and effort to do good work, and consequently botch the job. They then try to fob off their junk on unsuspecting users (authors, editors, and readers). Without copyright, the original designer cannot require the reproducer of a type to do a good job of reproduction. Hence, type quality is degraded by unauthorized copying.

Secondly, without protection, designs may be freely imitated; the plagiarist robs the original designer of financial compensation for the work. This discourages creative designers from entering and working in the field. As the needs of typography change (on-line documents and laser printing are examples of technical and conceptual changes) new kinds of typefaces are required. Creative design in response to such needs cannot flourish without some kind of encouragement for the creators. In a capitalist society, the common method is property rights and profit. In a socialist (or, in the past, royalist) society, the state itself might employ type artists. France, as a monarchy and as a republic has had occasional state sponsorship of typeface design over the past 400 years. The Soviet Union has sponsored the design of new typefaces, not only in the Cyrillic alphabet, but also in the other exotic scripts used by various national groups in the Soviet Union.

Those who would justify plagiarism often claim that the type artists do not usually receive a fair share of royalties anyway, since they have usually sold their designs to some large, exploitive corporation. It is true that type designers, like many artists, are often exploited by their publishers, but plagiarism exacerbates the problem. Plagiarism deprives the designer of decent revenues because it diverts profits to those who merely copied the designs. Plagiarism gives the manufacturer yet another excuse to reduce the basic royalty or other fee paid for typeface designs; the theme song is that the market determines the value of the design and cheap rip-offs debase the market value of a face. For those interested in the economic effects of piracy, it is clear that plagiarism of type designs ultimately hurts individual artists far more than it hurts impersonal corporations.

1.13.4 Kinds of Protection for Type

There are five main forms of protection for typefaces:

- * Trademark
- * Copyright
- * Patent
- * Trade Secret
- * Ethics

*** Trademark**

A trademark protects the name of a typeface. In the U.S., most trademarks are registered with the U.S. Patent and Trademark Office. The R in a circle (R) after a trademark or tradename indicates U.S. registration. The similarly placed TM indicates that a trademark is claimed, even if not yet officially registered. However, a trademark may be achieved through use and practice, even without registration. Owners of trademarks maintain ownership by use of the trademark and by litigation to prevent infringement or unauthorized use of the trademark by others.

As a few examples of registered typeface trademarks, there are Times Roman (U.S. registration 417,439, October 30, 1945 to Eltra Corporation, now part of Allied); Helvetica (U.S. Registration 825,989, March 21, 1967, also to Eltra-Allied), and Lucida (U.S. reg. 1,314,574 to Bigelow & Holmes). Most countries offer trademark registration and protection, and it is common for a typeface name to be registered in many countries. In some cases the registrant may be different than the originator. For example, The Times New Roman (Times Roman) was originally produced by the English

Monotype Corporation. In England and Europe, most typographers consider the design to belong to Monotype but the trademark was registered by Linotype (Eltra-Allied) in the U.S., as noted above.

Trademark protection does not protect the design, only the name. Therefore, a plagiarism of a design is usually christened with a pseudonym that in some way resembles or suggests the original trademark, without actually infringing on it. Resemblance without infringement can be a fine distinction.

Some pseudonyms for Times Roman are: English Times, London, Press Roman, Tms Rmn. Some for Helvetica are Helios, Geneva, Megaron, Triumvirate. So far, there seem to be none for Lucida. There are generic typeface classifications used by typographers and type historians to discuss styles, trends, and categories of design. Occasionally these apparently innocuous classification systems are employed by plagiarists to devise generic pseudonyms, such as Swiss 721 for Helvetica, and Dutch 801 for Times Roman. It is not certain whether this usage of a generic classification is more for clarification or for obfuscation. In general, the proper tradename is a better indicator of identity, quality, and provenience in typefaces than a generic name. Some people believe that the same is true for other commodities such as wine, where taste is important.

A trademark usually consists of both a proprietary and a generic part. For example, in the name Lucida Bold Italic, Lucida is the proprietary trademark part and Bold Italic is the generic part. The generic word type is usually understood to be a part of the name, e.g. Lucida Bold Italic type. Sometimes a firm will append its name or a trademarked abbreviation of it to the typeface name, to achieve a greater degree of proprietary content, e.g. B & H Lucida Bold Italic.

A related matter is the use of the name of a type's designer. A firm that ethically licenses a typeface will often cite the name of the designer — e.g. Stanley Morison (with Victor Lardent) for Times Roman, Max Miedinger (with Edouard Hoffmann) for Helvetica, Charles Bigelow and Kris Holmes for Lucida. Although a person's name is not usually a registered trademark, there are common law restrictions on its use. The marketing of plagiarized type designs generally omits the names of the designers.

Although trademark is an incomplete kind of protection, it is used effectively (within its limitations) to prevent the theft of type names. Certain traditional typeface names, usually the surnames of illustrious designers like Garamond, Caslon, Baskerville, Bodoni, and others have become generic names in the public domain. Trademark protection of such names requires the addition of some proprietary word(s), as with these hypothetical creations, Acme New Garamond, or Typoluxe Meta-Baskerville.

*** Copyright**

Copyright of typefaces can be divided into two parts: copyright of the design itself; and copyright of the font in which the design is implemented. In the U.S., typeface designs are currently not covered by copyright. This is a result of reluctance by the copyright office to deal with a complex field; by lobbying against copyright by certain manufacturers whose profits were based on typeface plagiarism; and by a reluctance of congress to deal with the complex issue in the recent revision of the copyright law.

The reluctance of Americans to press for typeface copyright may have been influenced by a feeling that typeface plagiarism was good for U.S. high-tech businesses who were inventing new technologies for printing, and plagiarizing types of foreign origin (Europe and England). If the situation becomes reversed, and foreign competition (from Japan, Taiwan, and Korea) threatens to overcome American technological superiority in the laser printer industry, then American firms may do an about-face and seek the protection of typeface copyright to help protect the domestic printer industry. Such a trend may already be seen in the licensing of typeface trademarks by Adobe, Hewlett Packard, IBM, Imagen, and Xerox in the U.S. laser printer industry.

In Germany, where typeface design has always been a significant part of the cultural heritage, and where typefounding has remained an important business, there are more than one kind of copyright-like protections for typefaces. Certain long-standing industrial design protection laws have been used to protect typeface designs in litigation over royalties and plagiarisms.

Further, there is a recent law, the so-called Schriftzeichengesetz enacted in 1981, that specifically protects typeface designs. New designs are registered, as is done with copyright in most countries. This law only protects new, original designs. It is available to non-German designers and firms. Therefore, some type firms and designers routinely copyright new designs in West Germany. This gives a degree of protection for products marketed in Germany. Since multinational corporations may find it cheaper to license a design for world-wide use rather than deal with a special case in one country, the German law does encourage licensing on a broader scale than would initially seem to be the case.

France, like Germany, has ratified an international treaty for protection of typefaces. This 1973 Vienna treaty will become international law when four nations ratify it. So far, only France and West Germany have done so, and thus a design must be protected separately in each country. Even when the treaty becomes law, it will take effect only in those countries that have ratified it. The treaty was principally the work of the late Charles Peignot, a French typefounder, and John Dreyfus, an English typographer and typographic scholar. Presently, typefaces may be registered for protection in France under a 19th century industrial design protection law.

In the U.S., there continues to be some movement for typeface design protection. A proposed bill that would protect the designs of useful articles, like type, has been in committee for several years. It seems to be going nowhere.

Digital (as opposed to analog) fonts may be protected by copyright of digital data and of computer programs. It has been established that computer software is copyrightable. Therefore, software that embodies a typeface, e.g. a digital font, is presumably also protected. There is some objection to this kind of copyright, on the grounds that the ultimate output of the program or the result of the data (i.e. a typeface design) is not copyrightable. However, the current belief expressed by the National Commission on New Technological Use of Copyrighted Works is that software is copyrightable even if its function is to produce ultimately a non-copyrightable work. Hence, typefaces produced by Metafont or PostScript(R), two computer languages which represent fonts as programs, are presumably copyrightable. Typefaces represented as bit-map data, run-length codes, spline outlines, and other digital data formats, may also be copyrightable. Some firms do copyright digital fonts as digital data.

Note that the designs themselves are still not protected in the U.S. A plagiarist could print out large sized letters (say, one per page) on an Apple LaserWriter, using a copyrighted PostScript digital font, and then redigitize those letters by using a scanner or a font digitizing program and thus produce a new digital font without having copied the program or digital data, and thus without infringing the copyright on the font. The quality of the imitation font would probably be awful, but it wouldn't violate copyright. Of course, the plagiarist would need to rename the font to evade trademark infringement. (As I write these words, I have the guilty feeling that I have just provided a recipe for type rip-off, but others have obviously thought of just such a scheme — John Dvorak has even proposed something like it in one of his columns.)

*** Design Patent**

The designs of typefaces may be patented in the U.S. under existing design patent law. Many designs are patented, but type designers generally don't like the patent process because it is slow, expensive, and uncertain. Nevertheless, some type do get patented, and it is a form of potential protection. Note that this is Design Patent — the typeface doesn't have to be a gizmo that does something, it merely has to be unlike any previous typeface. The drawback here is that most attorneys and judges are not aware that there are more than two or three typefaces: say, handwriting, printing, and maybe blackletter. Therefore, litigating against infringement is an educational as well as a legal process. It is easy to see that typeface theft is more subtle than knocking over a liquor store; it may not be illegal and the returns may be greater.

Protections like design patent are available in many other countries, but there is not an international standard (to my knowledge) so the situation must be examined on a country by country basis.

*** Invention Patent**

Methods of rendering typefaces can be patented as mechanical or electronic inventions. For example, the old hot-metal Linotype machinery was protected by various patents, as was the IBM Selectric typewriter and type ball. IBM neglected to trademark the typeface names like Courier and Prestige, so once the patents had elapsed, the names gradually fell into the public domain without IBM doing anything about it (at the time, and for a dozen years or so, IBM was distracted by a major U.S. anti-trust suit). Most students of the type protection field believe that those names are probably unprotectable by now, though IBM could still presumably make a try for it if sufficiently motivated.

There is currently a noteworthy development regarding a patent for outline representation of digital type as arcs and vectors, with special hardware for decoding into rasters. This patent (U.S. 4,029,947, June 14, 1977; reissue 30,679, July 14, 1981) is usually called the Evans & Caswell patent, after its inventors. It was originally assigned to Rockwell, and in 1982, Rockwell sued Allied Linotype for infringement. Allied settled out of court, having paid an amount rumored to be in the millions. Rockwell sold the patent, along with other typographic technology, to Information Internation, Inc. (III) which then sued Compugraphic for infringement. According to the Seybold Report, a respected typographic industry journal, Compugraphic recently settled out of court for 5 million. Although many experts believe the patent to be invalid because of several prior inventions similar in concept, it nevertheless seems to be a money-maker in corporate litigation. The Seybold Report has speculated on which firms III would litigate against next. Among the candidates suggested by the Seybolds was Apple for its LaserWriter, which uses outline fonts. Since the entire laser printer industry and the typesetting industry is moving toward outline font representation, Apple is certainly not alone. The Seybolds further speculate on whether the difference between character-by-character CRT typesetting and raster-scan laser typesetting and printing would be legally significant in such as case. Ultimately, some firm will hold out for a court judgement, and the matter will be decided.

*** Trade Secret**

Given that typeface designs have relatively little copyright protection in the U.S., they are often handled as trade secrets. The secret must apply to the digital data or programs only, because the images themselves are ultimately revealed to the public as printed forms. It is much more difficult to reconstruct the formula of Coca-Cola from its taste than it is to reconstruct the design of Helvetica from its look on the page. The exact bitmap or spline outline of a digital font is usually not reconstructable from the printed image, although CRT screen fonts at usual resolutions (60—120 dots per inch) may be reconstructed by patient counting and mapping of bits off a screen display. Typeface licenses often contain stipulations that the digital data will be encrypted and confidential. Just as a firm will protect the secret of a soft drink recipe, so a type firm will protect the exact nature of its digital data.

*** Ethics**

Some typographers are motivated by higher principles than greed, profit, expediency, and personal interest. Idealists enthused with concepts of ethical behavior and a vision of typography as a noble art may find it distasteful to use plagiarized types. Some graphic designers insist on using typefaces with bona-fide trademarks, both to ensure that the type will be of high quality, and to encourage creativity and ethics in the profession. A consequence of plagiarism that is sometimes overlooked is a general erosion of ethics in an industry. If it is okay to steal typeface designs, then it may be okay to purloin other kinds of data, to falsify one's resume, to misrepresent a product, and so forth. Most professional design organizations attempt to promote ethical standards of professional behavior, and personal standards may extend to avoidance of plagiarisms.

ATypI

The Association Typographique Internationale (ATypI) is an international organization of type designers, type manufacturers, and letterform educators. Its purpose is to promote ethical behavior in the industry, advancement of typographic education, communication among designers, and other lofty aims. Members of ATypI agree to abide by a moral code that restricts plagiarism and other forms of depraved behavior (pertaining to typography). These are noble goals, but some members (especially corporate members) of ATypI, confronted with the pressures and opportunities of commercial reality, nevertheless plagiarize typefaces of fellow members, the moral code notwithstanding. Since ATypI is a voluntary organization, there is very little that can be done about most such plagiarism. Some years back, a world-famous type designer resigned from the ATypI Board of Directors in protest over the organization's flaccid attitude toward the plagiarists among its ranks. He has since agreed to sit on the board again, but criticism of the organization's inability to prevent type rip-offs by its own members, not to mention by non-members, continues to be heard. Moderates in ATypI believe that a few morals are better than none. It is not clear whether their philosophical stance derives from Plato, Hobbes, or Rousseau. Given the general attitude of the public toward copyrighted video and software, it is doubtful that ethical considerations will hinder most end- users' attitude to plagiarized type fonts. A desire to have the fashionable label or trademark may be a greater motivation toward the use of bona-fide fonts than an ethical consideration.

Further Reading
The State of the Art in Typeface Design Protection, Edward Gottschall, Visible Language, Vol. XIX, No. 1, 1985. (A special issue on The Computer and the Hand in Type Design — proceedings of a conference held at Stanford University in August, 1983).

Der Schutz Typographischer Schriftzeichen, by Guenter Kelbel. Carl Heymans Verlag KG, Cologne, 1984. (A learned account in juridical German prose, of the significance of the Vienna Treaty of 1973 and the West German Schriftzeichengesetz of 1981).

Disclaimer
These notes were originally prepared at the request of Brian Reid for informal distribution. They are based on the author's review of available literature on the subject of typeface protection, and on personal experience in registering types for trademark, copyright, and patent. However, they are not legal advice. If one is contemplating protecting or plagiarizing a typeface, and seeks legal opinion, it is advisable to consult an attorney. The term plagiarize and words derived from it are used here in its dictionary sense of to take and use as one's own the ideas of another and does not mean that the practice of typeface plagiarism is illegal; that is determined by the laws of a particular country. Charles Bigelow is a professor of digital typography at Stanford University and a professional designer of original digital typefaces for electronic printers and computer workstations. Mr. Bigelow and his partner Kris Holmes designed the Lucida typeface family which is now widely used on various laser printers.

1.14. File Formats

Many different kinds of files are available on the net. These files contain many different kinds of data for many different architectures. Frequently, the extension (trailing end) of a filename gives a good clue as to the format of its contents and the architecture that it was created on.

In order to save space, most files on the net are compressed in one way or another. Many compression/decompression programs exist on multiple architectures. Multiple files and directories are often combined into a single 'archive' file. Many archive formats perform compression automatically.

1.14.1. File Format Extensions

*** .tar**
Unix 'tape archive' format. Tar files can contain multiple files and directories. Unlike most archiving programs, tar files are held together in a wrapper but are not automatically compressed by tar.

*** .Z**
Unix 'compress' format. Compression doesn't form a wrapper around multiple files, it simply compresses a single file. As a result, you will frequently see files with the extension .tar.Z. This implies that the files are compressed tar archives.

*** .hqx**
Macintosh 'BinHex' format. In order to reliably transfer Mac files from one architecture to another, they are BinHex encoded. This is actually an ascii file containing mostly hexadecimal digits. It is neither a compression program nor an archive wrapper.

*** .sit**
Macintosh 'Stuffit' archive.

*** .cpt**
Macintosh 'Compactor' archive. Like the .tar.Z format that is common among Unix archives, Macintosh archives frequently have the extensions .sit .hqx or .cpt.hqx indicating a BinHex'ed archive.

*** .arc**
PC 'arc' archive. This is an older standard (in PC terms, at least) and has gone out of fashion.

*** .zip**
PC 'zip' archive. This is the most common PC archive format today.

*** .arj**
PC 'arj' archive.

*** .zoo**
PC 'zoo' archive

*** .lzh**
PC 'lha/lharc' archive.

1.14.2. Font Formats

Just as the are many, many archive formats, there are many different font formats. The characteristics of some of these formats are discussed below. Once again, the file extension may help you to determine the font type. (On the Mac, the resource TYPE field is (probably) a better indicator).

*** PostScript Type 1 Fonts:**
Postscript Type 1 fonts (Also called ATM (Adobe Type Manager) fonts, Type 1, and outline fonts) contains information, in outline form, that allows a postscript printer, or ATM to generate fonts of any size. Most also contain hinting information which allows fonts to be rendered more readable at lower resolutions and small type sizes.

*** PostScript Type 3 Fonts:**
Postscript type 3 fonts are an old outline font format that is not compatible with ATM. Most developers have stopped using this format except in a few special cases, where special type 3 characteristics (pattern fills inside outlines, for example) have been used.

*** TrueType Fonts:**
Truetype fonts are a new font format developed by Microsoft with Apple. The rendering engine for this font is built into system 7 and an init, the Truetype init, is available for system 6 (freeware from Apple). It is also built into MS Windows v3.1.

Like PostScript Type 1 and Type 3 fonts, it is also an outline font format that allows both the screen, and printers, to scale fonts to display them in any size.

*** Bitmap Fonts:**
Bitmap fonts contain bitmaps of fonts in them. This a picture of the font at a specific size that has been optimized to look good at that size. It cannot be scaled bigger without making it look horrendously ugly. On the Macintosh, bitmap fonts also contain the kerning information for a font and must be installed with both type 1 and type 3 fonts. Their presence also speeds the display of commonly used font sizes.

1.14.3. Font Format Extensions

*** .afm**
Adobe Type 1 metric information in 'ascii' format (human parsable)

*** .bco**
Bitstream compressed outline

*** .bdf**
Adobe's Bitmap Distribution Format. This format can be converted to the platform specific binary files required by the local X Windows server. This is a bitmap font format distributed in ASCII.

*** .bez**
Bezier outline information

*** .chr**
Borland stroked font file

*** .fot**
MS-Windows TrueType format fonts

*** .gf**
Generic font (the output of TeX's MetaFont program (possibly others?))

*** .fli**
Font libraries produced by emTeX fontlib program. Used by emTeX drivers and newer

versions of dvips.

* .mf
TeX MetaFont font file (text file of MetaFont commands)

* .pfa
Adobe Type 1 Postscript font in ASCII format (PC/Unix) I believe that this format is suitable for directly downloading to your PostScript printer (someone correct me if I'm wrong ;-)

* .pfb
Adobe Type 1 PostScript font in "binary" format (PC/Unix)
Note: this format is not suitable for downloading directly to your PostScript printer. There are utilities for conversion between PFB and PFA.

* .pfm
Printer font metric information in Windows format

* .pk
TeX packed bitmap font file (also seen as . pk where is a number)

* .pl
TeX 'property list' file (a human readable version of .tfm)

* .ps
Frequently, any PostScript file. With respect to fonts, probably a Type3 font. This designation is much less 'standard' than the others. Other non-standard extensions are .pso, .fon, and .psf (they are a mixture of type 1 and type 3 fonts).

* .pxl
TeX pixel bitmap font file (obsolete, replaced by .pk)

* .sfl
LaserJet bitmapped softfont, landscape orientation

* .sfp
LaserJet bitmapped softfont, portrait orientation

* .sfs
LaserJet scalable softfont

* .tfm
TeX font metric file

* .vf
TeX virtual font which allows building of composite fonts (a character can be composed of any sequence of movements, characters (possibly from multiple fonts) rules and TeX specials)

* .vpl
TeX 'property list' (human readable) format of a .vf

1.15. Ligatures

A ligature occurs where two or more letterforms are written or printed as a unit. Generally, ligatures replace characters that occur next to each other when they share common components. Ligatures are a subset of a more general class of figures called "contextual forms."

Contextual forms describe the case where the particular shape of a letter depends on its context (surrounding letters, whether or not it's at the end of a line, etc.).

One of the most common ligatures is "fi". Since the dot above a lowercase 'I' interferes with the loop on the lowercase 'F', when 'f' and 'i' are printed next to each other, they are combined into a single figure with the dot absorbed into the 'f'. An example of a more general contextual form is the greek lowercase sigma. When typesetting greek, the selection of which 'sigma' to use is determined by whether or not the letter occurs at the end of the word (i.e., the final position in the word).

Amanda Walker provides the following discussion of ligatures:
Ligatures were originally used by medieval scribes to conserve space and increase writing speed. A 14th century manuscript, for example, will include hundreds of ligatures (this is also where "accents" came from). Early typefaces used ligatures in order to emulate the appearance of hand-lettered manuscripts. As typesetting became more automated, most of these ligatures fell out of common use. It is only recently that computer based typesetting has encouraged people to start using them again (although 'fine art' printers have used them all along). Generally, ligatures work best in typefaces which are derived from calligraphic letterforms. Also useful are contextual forms, such as swash capitals, terminal characters, and so on.

A good example of a computer typeface with a rich set of ligatures is Adobe Caslon (including Adobe Caslon Expert). It includes: Upper case, lower case, small caps, lining numerals, oldstyle numerals, vulgar fractions, superior and inferior numerals, swash italic caps, ornaments, long s, and the following ligatures:

ff fi fl ffi ffl Rp ct st Sh Si Sl SS St

(where S=long s)
[Ed: Another common example is the Computer Modern Roman typeface that is provided with TeX. this family of fonts include the ff, fi, fl, ffi, and ffl ligatures which TeX automatically uses when it finds these letters juxtaposed in the text.]

While there are a large number number of possible ligatures, generally only the most common ones are actually provided. In part, this is because the presence of too many alternate forms starts reducing legibility. A case in point is Luxeuil Miniscule, a highly-ligatured medieval document hand which is completely illegible to the untrained eye (and none too legible to the trained eye, either :)).

Don Hosek offers the following insight into ligatures:
Ligatures were used in lead type, originally in imitation of calligraphic actions (particularly in Greek which retained an excessive number of ligatures in printed material as late as the 19th century), but as typefaces developed, ligatures were retained to improve the appearance of certain letter combinations. In some cases, it was used to allow certain letter combinations to be more closely spaced (e.g., "To" or "Vo") and were referred to as "logotypes". In other cases, the designs of two letters were merged to keep the overall spacing of words uniform. Ligatures are provided in most contemporary fonts for exactly this reason.

Liam Quim makes the following observations:
The term ligature should only be used to describe joined letters in printing, not letters that overlap in manuscripts. Many (not all) accents came from the practice of using a tilde or other mark to represent an omitted letter, so that for example the Latin word 'Dominus' would be written dns, with a tilde or bar over the n. This is an abbreviation, not a ligature.

Most ligatures vanished during the 15th and 16th Centuries. It was simply too much work to use them, and it increased the price of book production too much.
[Ed: there is no "complete" set of ligatures.]

1.16. Standard Laser Printer Fonts

*** Postscript printers with 17 fonts have:**

Courier Courier-Bold Courier-BoldOblique
Courier-Oblique Helvetica Helvetica-Bold
Helvetica-BoldOblique Helvetica-Narrow Helvetica-Narrow-Bold
Helvetica-Narrow-BoldOblique Helvetica-Narrow-Oblique Helvetica-Oblique
Symbol Times-Bold Times-BoldItalic
Times-Italic Times-Roman

*** Postscript printers with 35 fonts have:**

All of the above, plus the following:
ZapfChancery-MediumItalic ZapfDingbats AvantGarde-Book
AvantGarde-BookOblique AvantGarde-Demi AvantGarde-DemiOblique
Bookman-Demi Bookman-DemiItalic Bookman-Light
Bookman-LightItalic NewCenturySchlbk-Bold
NewCenturySchlbk-BoldItalic
NewCenturySchlbk-Italic NewCenturySchlbk-Roman
Palatino-Bold Palatino-BoldItalic Palatino-Italic Palatino-Roman

*** HP LaserJet printers (II, IIP)**

Courier 10 Courier 12 LinePrinter 16.66

*** HP LaserJet printers (III, IIIP)**

All of the above, plus the following:
Scalable Times Roman and Scalable Univers using Compugraphic's Intellifont hinted font format.

*** SPARCPrinters**

The basic 35 fonts plus four scaled faces of each of Bembo, Gill Sans, Rockwell, Lucida, Lucida Bright, Sans and Typewriter, giving a total of 57 fonts, all in the F3 format.

1.17. Glossary

[I ripped this right out of the manual I wrote for Sfware. If you have comments, improvements, suggestions, please tell me...]

baseline

The baseline is an imaginary line upon which each character rests. Characters that appear next to each other are (usually) lined up so that their baselines are on the same level. Some characters extend below the baseline ("g" and "j", for example) but most rest on it.

bitmap

A bitmap is an array of dots. If you imagine a sheet of graph paper with some squares colored in, a bitmap is a compact way of representing to the computer which squares are colored and which are not.

In a bitmapped font, every character is represented as a pattern of dots in a bitmap. The dots are so small (300 or more dots-per-inch, usually) that they are indistinguishable on the printed page.

character

(1) The smallest component of written language that has semantic value. Character refers to the abstract idea, rather than a specific shape (see also glyph), though in code tables some form of visual representation is essential for the reader's understanding.
(2) The basic unit of encoding for the Unicode character encoding, 16 bits of information.
(3) Synonym for "code element".
(4) The English name for the ideographic written elements of Chinese origin.

download

Downloading is the process of transferring information from one device to another. This transferral is called downloading when the transfer flows from a device of (relatively) more power to one of (relatively) less power. Sending new fonts to your printer so that it "learns" how to print characters in that font is called downloading.

font

A particular collection of characters of a typeface with unique parameters in the 'Variation vector', a particular instance of values for orientation, size, posture, weight, etc., values. The word font or fount is derived from the word foundry, where, originally, type was cast. It has come to mean the vehicle which holds the typeface character collection. A font can be metal, photographic film, or electronic media (cartridge, tape, disk).

glyph

(1) The actual shape (bit pattern, outline) of a character image. For example, an italic 'a' and a roman 'a' are two different glyphs representing the same underlying character. In this strict sense, any two images which differ in shape constitute different glyphs. In this usage, "glyph" is a synonym for "character image'', or simply ``image''.
(2) A kind of idealized surface form derived from some combination of underlying characters in some specific context, rather than an actual character image. In this broad usage, two images would constitute the same glyph whenever they have essentially the same topology (as in oblique 'a' and roman 'a'), but different glyphs when one is written with a hooked top and the other without (the way one prints an 'a' by hand). In this usage, "glyph" is a synonym for "glyph type," where glyph is defined as in sense 1.

hints

When a character is described in outline format the outline has unlimited resolution. If you make it ten times as big, it is just as accurate as if it were ten times as small. However, to be of use, we must transfer the character outline to a sheet of paper through a device called a raster image processor (RIP). The RIP builds the image of the character out of lots of little squares called picture elements (pixels).

The problem is, a pixel has physical size and can be printed only as either black or white. Look at a sheet of graph paper. Rows and columns of little squares (think: pixels). Draw a large 'O' in the middle of the graph paper. Darken in all the squares touched by the O. Do the darkened squares form a letter that looks like the O you drew? This is the problem with low resolution (300 dpi). Which pixels do you turn on and which do you leave off to most accurately reproduce the character?

All methods of hinting strive to fit (map) the outline of a character onto the pixel grid and produce the most pleasing/ recognizable character no matter how coarse the grid is.

kerning

(noun): That portion of a letter which extends beyond its width, that is, the letter shapes that overhang - the projection of a character beyond its side-bearings.
(verb): To adjust the intercharacter spacing in character groups (words) to improve their appearance. Some letter combinations ("AV" and "To", for example) appear farther apart than others because of the shapes of the individual letters.

Many sophisticated word processors move these letter combinations closer together automatically.

outline font/format

See 'scalable font'

scalable font

A scalable font, unlike a bitmapped font, is defined mathematically and can be rendered at any requested size (within reason).

softfont

A softfont is a bitmapped or scalable description of a typeface or font. They can be downloaded to your printer and used just like any other printer font. Unlike built-in and cartridge fonts, softfonts use memory inside your printer. Downloading a lot of softfonts may reduce the printers ability to construct complex pages.

symbol set

The symbol set of a font describes the relative positions of individual characters within the font. Since there can only be 256 characters in most fonts, and there are well over 256 different characters used in professional document preparation, there needs to be some way to map characters into positions within the font. The symbol set serves this purpose. It identifies the "map" used to position characters within the font.

typeface

The features by which a character's design is recognized, hence the word face. Within the Latin language group of graphic shapes are the following forms: Uncial, Blackletter, Serif, Sans Serif, Scripts, and Decorative. Each form characterizes one or more designs. Example: Serif form contains four designs called Old Style, Transitional, Modern, and Slab Serif designs. The typeface called Bodoni is a Modern design, while Times Roman is a Transitional design.

1.18. Bibliography

Editors note: the following books have been suggested by readers of comp.fonts. They are listed in no particular order. I have lost the citations for some of the submissions. If you wrote a review that appears below and you aren't credited, please let norm know. I have decided that this is the best section for pointers to other font resources (specs and other documents, for example). These appear after the traditional bibliographic entries. As usual I will happily accept entries for this section. As of 9/92, the only files listed are the TrueType font information files available from Microsoft

Bill Ricker contributed the following general notes:

The Watson-Guptill, Godine, and Dover publishers all have many typography titles. Godine and Dover tend to be excellent; W-G tends toward 'how-to' books which are good for basics and juried Annuals of job work.

Hermann Zapf and his Design Philosophy, Society of Typographic Arts, Chicago, 1987.

On Stone, The Art and Use of Tyography on the Personal Computer, Sumner Stone, Bedford Arts, 1991.

Of the Just Shaping of Letters, Albrecht Durer, ISBN 0-486-21306-4. First published in 1525 as part of his theoretical treatise on applied geomentry.

"The Art of Measurment". Champ Flevry, Geofroy Troy.

First published in 1529 Troy attempts, in this book, to design an ideal Roman alphabet upon geometrical and aesthetic principles.

The Alphabet & Elements of Lettering, Frederic W. Goudy, ISBN 0-486-20792-7. Revised 1942 edition.

This very intresting book looks at the history of letter shapes as well font design.

The Mac is Not a Typewriter, Robin Williams, Peachpit Press.

A good, clear explanation of what typography is, and how to get it from your computer. Mac-specific, but full of excellent general advice. I think there's also a PC version. Available at most computer bookstores

Rhyme and Reason: A Typographic Novel, Erik Spiekermann, H. Berthold AG, ISBN 3-9800722-5-8. Printing Types (2 vols), Daniel Berkely Updike, Dover Press.

Affordable edition of the most readable history of type, lots of illustrations. Notes: it has been suggested that the Dover edition was 3 volumes, paperback, and the Harvard U. P. editions were 2 volumes hardback. It appears that the Dover edition is out of print. Collectible HUP editions are not cheap although later HUP editions may be had. Most libraries have later HUP and Dover editions. If someone knows of a source, please pass it along.

The Art of Hand Lettering, Helm Wotzkow, Dover Press, reprint from 1952.

Looking Good In Print, Roger C. Parker, Ventana Press, ISBN 0-940087-32-4.

Well, as a beginner's book, it isn't bad. I can't say that I agree with the author's tastes all the time, but he at least gives some good examples. Also there are some nice _Publish_-style makeovers.

Book Design: A Practical Introduction, Douglas Martin, Van Nostrand Reinhold, New York: 1989. 206pp.

Along with Jan White's book (see below), this provides a fairly complete guide to book design. Martin's book is somewhat more conservative in outlook and also reflects his UK background.

Digital Typography: An Introduction to Type and Composition for Computer System Design, Richard Rubinstein, Addison-Wesley, Reading, Massachusetts: 1988. 340pp.

An interesting, technological approach to typography which is worth reading although not necessarily always worth believing. A not insubstantial portion of the text is dedicated to representing type on a CRT display and Rubinstein devotes some time to expressing characteristics of typography numerically.

Graphic Design for the Electronic Age, Jan V. White, Watson-Guptill Publications, New York: 1988. 212pp.

A good handbook for document design. In a well-organized approach, White covers the principles for laying out most of the typographics features of a technical document. White is a bit overeager to embrace sans-serif types and in places his layout ideas seem a bit garish, but it's still a quite worthwhile book.

Xerox Publishing Standards: A Manual of Style and Design, Watson-Guptill Publications, New York: 1988. 400pp.

Overall, a disappointing book. It is divided into four sections of widely varying intent: "Publishing Process," "Document Organization,"``Writing and Style'' and ``Visual Design.'' None of them is really adequate for the task and all are highly centered on the Xerox method for publishing. As a guide to Xerox' process, it succeeds, but as a manual for general use, it falls far short. In print.

Methods of Book Design (3rd edition), Hugh Williamson, Yale University Press, New Haven: 1983. 408pp.

It is a bit out-of-date as regards technology, but on issues relating purely to design it is comprehensive and definitive. Well, I suppose it could be argued that printing technology influences design e.g. some types look fine in metal but lousy in digital imagesetting and therefore a book that is out-of-date in technology can't really be ``definitive'' in matters of design either. In any event, _Methods_ is more than adequate for a beginner's needs. My paper-bound copy (ISBN 0-300-03035-5) was $ 13.95; cheap at twice the price!

The Thames & Hudson Manual of typography, Rauri McLean, Thames & Hudson

An excellent book if you start getting more interested in type. Look for Rauri McLean's other books after this one...

Typography and Why it matters, Fernand Baudin.

There is no better introduction than it. It's not a primer on subjects such as "what does Avant Garde look like," or "This is a good font for books." It is a good primer on the things you need to know before the rest should be considered. He's a lovely writer, to boot.

[My copy is at work, so I may have munged the title—look up Baudin in "Books in Print" and improvise :-)]

Better Type, Betty Binns

It's definitely not a lightweight beginner's introduction, but I've found it to be indispensable. It's a large-format hardcover, but you can find it remaindered for cheap if you look around. The book goes into great detail about how factors like line spacing, line length, point size, and design of typeface (evenness of stroke weight, x-height, etc.) affect readability. When you've gotten the basics out of the way and want to learn more about the fine nuances of type color, this book is an absolute must.

Printing Types: An Introduction..., S. Lawson, (revised) 1990

I'd also recommend Alexander S. Lawson's books especially /Printing Types: An Intro.../(revised), 1990, which includes electronic types now.

Twentieth Century Type Designers, Sebastian Carter, 1987.

Discusses adaptaters of old faces to machine caster and film/laser, as well as new works.

Tally of Types, Stanley Morrison, Cambridge University Press.

A keepsake for CUP on the Monotype fonts he'd acquired for them when he was Type Advisor to both Brit.Monotype & CUP (Cambridge University Press, Cambs.UK), which discusses his hindsight on some of the great revival fonts and some of the better new fonts.

Chicago Manual of Style, University of Chicago Press, 1982; ISBN 0-226-10390-0.

The chapter on Design and Typography is most directly relevant, but there are a lot of hints scattered all through the Chicago Manual on making your words more readable and your pages more attractive.

X Window System Administrator's Guide (O'Reilly X Window System Guides, volume 8), O'Reilly

It gives advice about setting up fonts, etc.

How Bodoni intended his types to look Bodoni, Giambattista. Fregi e Majuscole Incise e Fuse de ... Bodoni, Harvard University Library (repr).

Inexpensive collectible, reproduced as a keepsake by the Houghton Library at Harvard.

The Elements of Typographic Style, Robert Bringhurst, Hartley & Marks 0-88179-033-8 pbk $ 15, Z246.B74 1992 0-88179-110-5 cloth, $25.

A typography for desktop publishers who want to absorb some style. Informed by the historical european tradition and the desktop advertising, tempered by oriental yin-yang and examples. A page-turner with repeat-read depth.

The only book I've seen that discusses page proportions that admits there are more than three ways that describes how to find one that feels good for your page. Hermann Zapf on the cover-blurb: "All desktop typographers should study this book. ... I wish to see this book become the Typographers' Bible.'

Printing It, Clifford Burke, Ballantine, 0-345-02694-2.

Manual for the hobby letterpress printer.

Twentieth Century Type Designers, Sebastian Carter, Taplinger, 1987.

Discusses the talented adaptators of old faces to machine caster and film/laser, as well as the designers of new works. Indexed?

Design with Type, Carl Dair, University of Toronto Press, 0-8020-1426-7.

In print again (or still?); the ISBN above may be stale. A great introduction to the issues of practicality and taste that confront the users of type. A prized possession. I only regret that the book does not include among the excerpts from his Westvaco pamphlets the Seven Don'ts of Typography.

Typography 6: The Annual of the Type Directors Club, Susan Davis, ed., Watson-Guptill, 0-8230-5540-x.

Specimens of Type Faces in the U.S. G.P.O., John J. Deviny, director., US G.P.O.

Practice of Typography: Plain Printing Types, Theodore Low De Vinne, Century Co./DeVinne Press.

One of the earlier critical studies, in four volumes of which this is my personal favorite, and still a classic reference. If one wants to understand 18th and 19th century typography in context, this writer lived the transition from eclectic to standard sizes, and comments with taste.

An Essay on Typography, Eric Gill, Godine, 0-87923-762-7.

The Alphabet and Elements of Lettering, Frederic W. Goudy, Dorset Press (Marboro Books), 0-88029-330-6

Lovely. A wonderful way to learn Goudy's taste.

Stanley Morison Displayed, Herbert Jones, Frederick Muller Ltd /W, 0-584-10352-2.

Lovely. A wonderful way to learn Morison's taste.

Printing Types: An Introduction..., Alexander S. Lawson et. al.,Beacon 1971,?Godine? 1990; (2nd Ed includes electronic types now)

"Good introduction to comparisons of typefaces, with a detailed history and a key family or face of each general category. Denounces rigid indexes of type faces."

Anatomy of a Typeface, Alexander Lawson, Godine, 0-87923-333-8, Z250.L34 1990

Deep description of the authors' favorite exemplar and its influences and relatives in each type category. It follows, without explicating, the category system developed in the prior book.

Types of Typefacs and how to recognize them, J. Ben Lieberman, Sterling, 1968

"This isn't very good really, but it does give lots of examples of the main categories." Old bibliographies praised this one, but I haven't seen it so I can't comment.

Tally of Types (& other titles), Stanley Morrison, Cambridge U. Press.

A keepsake for CUP on the Monotype fonts he'd acquired for them when he was Type Advisor to both Brit. Monotype & CUP (Cambridge University Press, Cambs.UK), which discusses his hindsight on some of the great revival fonts and some of the better new fonts.

Rookledge's International Type Finder 2nd, Perfect, Christopher and Gordon Rookledge, Ed Moyer Bell Ltd / Rizzoli, 1-55921-052-4,| Z250.P42 [1st Ed was NY: Beil 1983]

"Lg. trade pb. Indexed by stylistic & characteristic features. Shows A-Z, a-z, 0-9 in primary figures, whether lining or ranging. Particularly distinctive sorts are marked for ease of comparison. Separate tables collect the distinctive characters for assistance in identifying a sample."

English Printers' Ornaments, Henry R. Plomer, Burt Franklin

Paragraphs on Printing, Bruce Rogers, [Rudge] Dover, 0-486-23817-2

Digital Typography: An Introduction to Type and Composition for Computer System Design, Richard Rubinstein, Addison-Wesley, Reading, Massachusetts: 1988. 340pp.

For people who are disappointed with how the type looks on the laser, this book explains the subleties of that medium and of the screen that others miss. This is a study of the Human Factors of computer typographic systems.

The Case for Legibility, John Ryder, The Bodley Head, 0-370-30158-7, Z250.A4

The Solotype Catalog of 4,147 Display typefaces, Dan X. Solo, Dover, 0-486-27169-2, Z250.5.D57S654 19

"Working catalog of a specialty Graphics Arts shop. They use proprietary optical special effects techniques to get Desktop Publishing effects, and more, without the laser-printer grain. Great listing of 19th Century Decorated Types — probably the largest collection in the world. Prices to order headlines from them are NOT cheap however. Their services are for professional or serious hobby use only. Solo's previous Dover books show some number of complete alphabets of a general peculiar style; this one shows small fragments of his entire usable collection, important as an index. (According to private correspondence, they have more faces that have not yet been restored to usable condition.) Not well indexed, but indexed."

Stop Stealing Sheep & find out how type works, Erik Spiekermann & E.M. Ginger., Adobe Press, 1993

Introductory, motivational. If you wonder why there are so many type faces in the world, this is the book for you! The title refers to the old joke: "A man who would letterspace lowercase would also steal sheep."

The Art & Craft of Handmade Paper, Vance Studley, Dover, 0-486-26421-1, TS1109.S83 1990|

Letters of Credit, Walter Tracey, Godine Press

. "I can't recommend this too highly. It's not as introductory as the Sheep Book, but conveys a feeling of love and respect for the letter forms, and covers a lot of ground very, very well."

Printing Types: Their History, Forms & Use, Daniel Berkely Updike, Harvard University Press, reprint by Dover.

The standard reference. Tour-de-force history of type and type-styles. A trifle conservative in its biases, but typography is conservative for good reason: readibility. Check the addenda for his final words on newer faces.

1. I believe the Dover edition to be 3 vols Pbk; both the collectable and later Harvard U.P. editions were two vols hbk.

2. I am informed by my bookseller & Books In Print that the Dover edition is out of print. *sigh* If a source be known, let me know. Collectible HUP eds are not cheap, although later HUP eds may be had. Most libararies have later HUP or Dover eds.

Modern Encyclopedia of Typefaces, 1960-90, Lawrence W. Wallis, Van Nostrand Reinhold, 0-442-30809-4, Z250.W238 1990

"Gives examples of most typefaces, almost all digital, designed & distributed in the last 30 years. Cross indexed by foundry and designer, and sources and looks-likes. Some historical bits. Shows full a-z,A-Z,0-9, a few points (punctuation); and 0-9 again if both lining and oldstyle supplied. Only complaint is that it omits small caps even from what few fonts have 'em and the accented characters, of which most have some but too few. List $ 25."

About Alphabets: Some Marginal Notes on Type Design, Hermann Zapf, MIT Press, 0-262-74003-6

Hermann Zapf & His Design Philosophy, Hermann Zapf, Society of Typographic Arts, Chicago

"Anything about, by, or vaguely connected with Hermann Zapf is probably worth reading several times :-)"

Manuale Typographicum, Hermann Zapf, MIT Press, 0-262-74004-4

There are two books of this title (portrait and landscape); this is the only mass-market edition of either. Both are Zapf's selections of interesting typographical quotations in his inimitable display typography.

1.19. (En)Coding Standards

Unicode Consortium; The Unicode Standard, volumes 1 and 2, Worldwide Character Encoding, Addison-Wesley Publishing Co. Unicode consortium e-mail address is: < unicode-inc@hq.m4 .metaphor.com>

To obtain more information on Unicode or to order their printed material and/or diskettes Steven A. Greenfield Unicode Office Manager, 1965 Charleston Road Mountain View, CA 94043 Tel. 415-966-4189 Fax. 415-966-1637

Xerox Character Code Standard, Xerox Corp., Xerox Systems Institute, 475 Oakmead Parkway, Sunnyvale, CA 94086

1.20. TrueType

George Moore <georgem@microsoft .com> announces the following information regarding TrueType fonts:

"I am pleased to announce that there is now one central location for all official Microsoft TrueType information available on the Internet. The 9 files listed below are available for anonymous ftp access on ftp.uu.net (137.39.1.9) in the /vendor/microsoft/TrueType-Info directory. The most important of those files is the TrueType Font Files Specifications, a 400 page book which describes in excruciating detail how to build a TrueType font. Other information is also available in the same directory and other files will be added from time to time.

For those people who do not have ftp access to the Internet can find the same information available for downloading on Compuserve in the Microsoft developer relations forum (GO MSDR) in the TrueType library.

Please be aware that the TrueType specifications is a copyrighted work of Microsoft and Apple and can not be resold for profit.

TrueType developer information files on ftp.uu.net: 1. ttspec1.zip, ttspec2.zip, and ttspec3.zip

The TrueType Specification: These three compressed files contain the "TrueType Font Files Specifications", a 400 page book complete with illustrations which details how to construct a TrueType font from scratch (or build a tool to do so), the TrueType programming language, and the complete format of each sub-table contained in the .TIF file. These documents are stored in Word for Windows 2.0 format and require Windows 3.1 for printing. See the "readme.doc" (in ttspec1.zip) for printing instructions. Requires 2.5MB of disk space after uncompression. This manual is a superset of the similar specifications from Apple and has added information specific to Windows that is not present in the Apple version.

2. ttfdump.zip
An MS-DOS executable which will dump the contents of a TrueType font out in a human-readable fashion. It allows you to dump the entire font, or just specific sub-tables. This tool, combined with the specifications above, allows very effective debugging or exploration of any TrueType font. For example, to dump the contents of the 'cmap' (character code to glyph index mapping) table, enter:

```
ttfdump  fontname.ttf  -tcmap  -nx
```

Entering "ttfdump" with no options will give you a help message.

3. ttfname.zip
Example C source code on how to parse the contents of a TrueType font. Although this particular example will open up the file and locate the font name contained within the 'name' table, it could be readily adapted to parse any other structure in the file. This compressed zip file also contains many useful include files which have predefined structures set up for the internal tables of a TrueType font file. This code may be useful for developers who wish to parse the TrueType data stream returned by the GetFontData() API in Windows 3.1.

4. tt-win.zip
A 31 page Word for Windows 2.0 document which is targeted for the Windows developer who is interested in learning about some of the capabilities TrueType adds to Windows 3.1. Contains many illustrations.

5. embeddin.zip
A text file which describes all of the information necessary for a Windows developer to add TrueType font embedding capabilities to their application. Font embedding allows the application to bundle the TrueType fonts that were used in that document and transport it to another platform where the document can be viewed or printed correctly.

6. tt-talk.zip
The TrueType Technical Talks 1 and 2. These text files describe some of the things that are happening with TrueType behind the scenes in Windows 3.1. The first document walks the reader through all of the steps that occur from when the user first presses the key on the keyboard until that character appears on the screen (scaling, hinting, dropout control, caching and blitting). The second talk describes one of the unique features of TrueType called non-linear scaling which allows the font vendor to overcome some of the physical limitations of low resolution output devices.

7. lucida.zip
This text file contains useful typographic information on the 22 Lucida fonts which are contained in the Microsoft TrueType Font Pack for Windows. It gives pointers on line-layout, mixing and matching fonts in the family and a little history on each typeface. This information was written by the font's designers, Chuck Bigelow & Kris Holmes."

1.21. Rules of Thumb

It is difficult to set out guidelines for font usage, because almost any rule can be brilliantly broken under the right circumstances.

General guidelines:

• Never lose track of the kind of work you're doing. An effect that would ruin a newsletter might be just the thing for a record cover. Know when you can safely sacrifice legibility for artistic effect.

• Keep in mind the final reproduction process you'll be using. Some effects (like reversed type, white on black) can be hard to read off an ordinary 300-dpi laser, but will work if finals are done on a high-resolution printer, such as a Linotronic. Will the pages be photocopied? Offset? Onto rough paper, shiny paper? All these factors can and should influence your choice of fonts and how you use them.

• Running some comparative tests is a good idea. Better to blow off a few sheets of laser paper now than to see a problem after thousands of copies are made.

• No one can teach you font aesthetics; it must be learned by example. Look at beautiful magazines, posters, books with wide eyes, so that you can see how it's done. Examine ugly printed matter critically and consider why it's hard to read.

Good rules of thumb:

• If you need a condensed font, find one that was designed that way, rather than scaling an existing font down to a percentage. Any scaling distorts a font's design; excessive scaling interferes with legibility - this goes for widening as well as narrowing. Extended faces do exist, although they aren't as common as condensed ones.

• Many people feel that bold or italic type, or type in ALL CAPS, is more legible: "This is the most important part of the newsletter, let's put it in bold." In fact, legibility studies show that such type is actually harder to read in bulk. Keep the text in a normal style and weight, and find another way to emphasize it - box it, illustrate it, run it in color, position it focally.

• Too much reverse type - white on black - is hard on the eyes. It can be a nice effect if used sparingly. Don't reverse a serif font, though - its details will tend to fill in. Stick to reversing bold sans-serifs, and remember to space them out a bit more than usual.

• It is always safest to use a plain serif font for large amounts of text. Because Times is widely used, it doesn't mean it should be avoided. Fonts like Palatino, Times, Century Old Style are deservedly popular because people can read a lot of text set in such faces without strain.

• Don't expect anyone to read extensive text set in a condensed font.

• As point size gets bigger, track tighter, and (if the software allows) reduce the spacebands as well. A spaceband in a headline size (anything over 14 point) should be about as wide as a letter "i".

• If you only have a few large headlines, hand-kerning the type, pair by pair, can make the end result much more pleasing. Besides, working with fonts this closely makes them familiar.

• Column width and justification are major elements in design. The narrower the column, the smaller the type can be; wide rows of small type are very hard to read. Often it's a better idea to set narrow columns flush left rather than justified, otherwise large gaps can fall where hyphenation isn't possible.

• Use curly quotes.

• Don't put two spaces at the end of a line (.) instead of (.) when using a proportionally spaced font.

1.23. A Brief Introduction to Typography

Space, time, and bandwidth are too limiting to provide a complete introduction to typography in this space. I'd be very willing to make one available for anonymous ftp, if you want to write one, but I'm not going to write it—I have neither the time nor the expertise. However, the following description of Times, Helvetica, and Courier will suffice for a start. For more information, several books on typography are listed in the bibliography.

1.23.1. Comments by Laurence Penney:

Laurence Penney offers the following description of Times, Helvetica, and Courier:

Times is a typeface designed in the 1930s for the Times newspaper in London and is now used widely in books, magazines and DTP. Its design is based on the typographical principles evolved since Roman times (upper case) and the 16th century (lower case). It is called a TRANSITIONAL typeface, after the typefaces of the 17th century which it resembles. Like all typefaces designed for typesetting large quantities of text, it is proportionally spaced: the i takes about a third the width of an M. Personally I don't like Times too much and prefer the more elegant Garamond and Baskerville, but these will probably cost you money... Note: The Transitionals came after the Old Styles (like Garamond) and before the Moderns (like Bodoni).

Helvetica is an example of a SANS-SERIF typeface. These first appeared in the late 19th century in Germany and flourished in the 1920s and 30s, when they were regarded as the future of typography. It's more a geometric design than the humanist design of Gill Sans, but less geometric than Avant Garde and Futura. To my mind it lacks elegance, and Adrian Frutiger's Univers shows how this kind of typeface should be done. (Just compare the B, R, Q, a, g of Univers and Helvetica to see what I mean — and don't you just love Univers's superbly interpreted ampersand ?!) Helvetica is one of the few fonts that is improved by its BOLD version.

Another interesting approach to sans-serif is Optima, by Hermann Zapf, which keeps the stroke-weight variations which sans-serifs usually reject. Use sans-serif fonts for the same applications as Times, above, but where you're less concerned with elegance, and more with a functional appearance - they're generally reckoned to be slightly less legible than good serifed fonts. They're also very suitable for display work.

Courier is a typeface derived from typewriter styles. It should ONLY be used when you want to simulate this effect (e.g. when writing letters Courier usually appears "friendlier" than Times). Like all typewriter fonts, it is MONOSPACED (characters all have the same width) and is thus suitable for typesetting computer programs. However there are nicer looking monospace fonts than Courier (which has oversize serifs), that still remain distinct from the text fonts like Times and Helvetica. A good one is OCR-B, designed by Frutiger. Note that monospaced fonts are less economical on space than proportional fonts.

[ed: Following the original posting of this message, Laurence Penny and Jason Kim discussed the issue privately. The following summary of their discussion may serve to clarify some of the more subtle points. My thanks to Laurence and Jason for allowing me to include this in the FAQ.]

LP-1> The Transitionals came after the Old Styles (like Garamond) and before the Moderns (like Bodoni).

JK> Not necessarily true! Ideologically, yes, but not chronologically. I believe, for example, that Bodoni predates New Century Schoolbook or some such typeface.

LP-2> What I meant by "X came after Y" was "the first examples of X appeared after the first examples of Y" — it's called precis. Some people still make steam trains, but you can still say "Steam engines came before diesels." This is chronological, not ideological in my

LP-1> Another interesting approach to sans-serif is Optima, by Hermann Zapf, which keeps the stroke-weight variations which sans-serifs usually reject. Use sans-serif fonts for the same applications as Times, above, but where you're less concerned with elegance, and more with a functional appearance - they're generally reckoned to be slightly less legible than good seriffed fonts. They're also very suitable for display work.

JK> Slightly? I have several textbooks typeset by utter fools and they are a pain in the ass (and eyes) to read! Please don't encourage anyone to use Optima (or any sans serif fonts for that matter) "for the same applications as Times," which, need I remind you, was designed for *newspaper* work!!

LP-2> OK, maybe I was a little over-generous to Univers, Helvetica, etc., but I think variation is extremely important in typography. Have you ever read the British magazine "CAR" ? That uses Helvetica light (I think) in a very legible and attractive way, IMO. I agree, though, Optima is crappy for text, but it's a very valuable experiment and looks beautiful when printed in high quality for titling, etc. And yes, *books* in Helvtica are generally

JK> Serifs have been scientifically shown to be a *lot* easier on the reader, as they guide the eyes along the lines.

LP-2> In all tests I've seen the serifs have always won the day, but only with certain seriffed fonts, and fonts like Univers aren't far behind. The "tracking" advantage for serif fonts is reduced when you're talking about narrow newspaper/magazine

JK> You wrote a pretty short and partial history of type. Why ignore the roots of type (blackletter) as well as the climax (moderns—give an explanation) and subsequent 'post-modern' revivals?

LP-2> I was just talking about the place the 3 most common DTP types hold in the history of typography, and a few associated pitfalls. It wasn't meant as a ``history of typography'' at all. Please feel free to provide such a history yourself.

JK> I think any short list of specific faces is incomplete without mention of Palatino, the most popular Old Style revival in existence.

LP-2> Do you? To my mind Palatino is grossly overused. You must agree it looks bad for dense text. It isn't a proper ``oldstyle revival'' at all, more of a ``calligraphic interpretation'' of it. Zapf designed it as a display face, and wasn't too concerned about lining up the serifs (check out the "t"). And it just *has* to be printed on 1200dpi devices (at least) to look good in small sizes. OK then, maybe a short list is incomplete without a caution NOT to use Palatino...

JK> Also, if this is meant to be a "quick history/user guide for those fairly new to using fonts on desktop publishing systems," then I would recommend more directions about the proper uses of certain faces (e.g., Goudy for shaped text, Peignot for display *only*) and styles (e.g., italics for editorial comments, all-caps for basically nothing).

LP-2> Okay, okay. I was only sharing a few ideas, not trying to write a book. Surely you agree that the 3 typefaces I chose are by far the most commonly used and abused these days? I don't think a discussion of Goudy or Peignot fits in very well here, unless we're hoping to make a very wide-ranging FAQL.

Regarding styles: first, italics are used principally for *emphasis* (rather than bold in running text); second, all good books have a few small caps here and there, don't they? - all mine do...

JK> Sorry if I come across as critical. I think the idea of making a FAQL is a good one, as is your effort. We just have to make sure it doesn't give any newbies the wrong impressions and further perpetuate the typographical morass we're facing today.

LP-2> Sorry if I come across as defensive, but I stand by what I said and object to the suggestion that I am "perpetuating the typographical morass''. (I don't know if you really intended this - apologies if you didn't.)

1.23.2. Comments by Don Hosek:

Don Hosek offers the following additional notes:

The "Times" in most printers is actually a newer version of the font than Monotype's ``Times New Roman'' which it is originally based on. Walter Tracy's _Letters of Credit_ gives an excellent history of the face which was based on Plantin and in the original cutting has metrics matching the original face almost exactly. Another interesting note about the face is that it is almost a completely different design in the bold: this is due to the fact that old-styles are difficult to design as a bold. Incidentally, the classification of Times as a transitional is not firm. It likely is placed there by some type taxonomists (most notably Alexander Lawson) because of the bold and a few minor features. Others, myself included, think of it as a old style. The typeface listed in the Adobe catalog as Times Europa was a new face commissioned in 1974 to replace the old Times (whose 50th birthday was this past October 3rd). Hermann Zapf is not particularly pleased with any of the phototypesetting versions of Optima. As a lead face, Optima is very beautiful. His typeface "World", used in the World Book Encyclopedia is one recutting for photocomp which improves the font somewhat. He is on record as saying that if he had been asked, he would have designed a new font for the technology.

1.24. What does 'lorem ipsum dolor' mean?

'Lorem ipsum dolor' is the first part of a nonsense paragraph sometimes used to demonstrate a font. It has been well established that if you write anything as a sample, people will spend more time reading the copy than looking at the font. The "gibberish" below is sufficiently like ordinary text to demonstrate a font but doesn't distract the reader. Hopefully.

Lorem ipsum dolor sit amet, consectetaur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum Et harumd und lookum like Greek to me, dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda. Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam makes one wonder who would ever read this stuff? Bis nostrud exercitation ullam mmodo consequet. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint occae. Et harumd dereud facilis est er expedit distinct. Nam libe soluta nobis eligent optio est congue nihil impedit doming id Lorem ipsum dolor sit amet, consectetur adipiscing elit, set eiusmod tempor incidunt et labore et dolore magna aliquam. Ut enim ad minim veniam, quis nostrud exerc. Irure dolor in reprehend incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend. Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. At vver eos et accusam dignissum qui blandit est praesent. Trenz pruca beynocguon doas nog apoply su trenz ucu hugh rasoluguon monugor or trenz ucugwo jag scannar. Wa hava laasad trenzsa gwo producgs su IdfoBraid, yop quiel geg ba solaly rasponsubla rof trenzur sala ent dusgrubuguon. Offoctivo immoriatoly, hawrgasi pwicos asi sirucor.Thas sirutciun applios tyu thuso itoms ghuso pwicos gosi sirucor in mixent gosi sirucor ic mixent ples cak ontisi sowios uf Zerm hawr rwivos. Unte af phen neige pheings atoot Prexs eis phat eit sakem eit vory gast te Plok peish ba useing phen roxas. Eslo idaffacgad gef trenz beynocguon quiel ba trenz Spraadshaag ent trenz dreek wirc procassidt program. Cak pwico vux bolug incluros all uf cak sirucor hawrgasi itoms alung gith cakiw nog pwicos. Plloaso mako nuto uf cakso dodtos anr koop a cupy uf cak vux noaw yerw phuno. Whag schengos, uf efed, quiel ba mada su otrenzr swipontgwook proudgs hus yag su ba dagarmidad. Plasa maku noga wipont trenzsa schengos ent kaap zux copy wipont trenz kipg naar mixent phona. Cak pwico siructiun ruos nust apoply tyu cak UCU sisulutiun munityuw uw cak UCU-TGU jot scannow. Trens roxas eis ti Plokeing quert loppe eis yop prexs. Piy opher hawers, eit yaggles orn ti sumbloat alohe plok. Su havo loasor cakso tgu pwuructs tyu InfuBwain, ghu gill nug bo suloly sispunsiblo fuw cakiw salo anr ristwibutiun. Hei muk neme eis loppe. Treas em wankeing ontsime ploked peish rof phen sumbloat syug si phat phey gavet peish ta paat ein pheeir sumbloats. Aslu unaffoctor gef cak siructiun gill bo cak spiarshoot anet cak GurGanglo gur pwucossing pwutwam. Ghat dodtos, ig pany, gill bo maro tyu ucakw suftgasi pwuructs hod yot tyubo rotowminor. Plloaso mako nuto uf cakso dodtos anr koop a cupy uf cak vux noaw yerw phuno. Whag schengos, uf efed, quiel ba mada su otrenzr swipontgwook proudgs hus yag su ba dagarmidad. Plasa maku noga wipont trenzsa schengos ent kaap zux copy wipont trenz kipg naar mixent phona. Cak pwico siructiun ruos nust apoply tyu cak UCU sisulutiun munityuw uw cak UCU-TGU jot scannow. Trens roxas eis ti Plokeing quert loppe eis yop prexs. Piy opher hawers, eit yaggles orn ti sumbloat alohe plok. Su havo loasor cakso tgu pwuructs tyu.

[This version was found on CompuServe. It differs from other versions I have seen in print, increasingly so as you go along. It almost looks computer-generated, doesn't it?]

Ytterligare en omgång "Vanliga frågor om teckensnitt"

MS-Dos Information

3.1. MS-DOS font notes

The easiest way to get outline fonts under MS-DOS is with MicroSoft Windows 3.x or OS/2 2.x.

MicroSoft Windows 3.0 with Adobe Type Manager (ATM) and OS/2 2.0 support PostScript Type1 fonts.

MicroSoft Windows 3.1 supports TrueType fonts natively.

Bitmap fonts are available in a variety of formats: most formats are designed with the printer in mind and not the display since (prior to graphical environments like Windows, GEM, and OS/2) the majority of work under MS-DOS was done with a character-based interface.

3.2. Frequently requested fonts

Many fonts are available at various archives. The biggest font archive for MS-DOS format fonts is ftp.cica.indiana.edu. Note: you can use any Mac format Type1 font on your PC by converting it to PC format with the free/shareware as described below.

The following fonts are in Type 1 format for MS-DOS. Some are also available in TrueType format.

* Hebrew ShalomScript
ShalomOldStyle, ShalomStick
* Japanese
Shorai
* Star Trek
Crillee, TNG monitors
* IBM OEM Line Drawing Characters

Try Adobe PrestigeElite or Adobe LetterGothic. They have all the characters you want, but the 'line draw' characters are unencoded—you will need tools to reencode the outline font itself and make a new PFM metric files.

Or try IBMExtended from Impramatur Systems in Cambridge, Mass. It already is encoded using IBM OEM encoding (some DOS code page).

The IBM version of Courier distributed freely under the X11 Consortium also contains the appropriate characters. Again, the font will have to be reencoded for Windows. Appropriate AFM files for this font can be obtained from: ibis.cs.umass .edu:/pub/norm/comp.fonts. The file is called IBM-Courier-PC8-SymbolSet-AFMs .zip.

3.3. Font Installation

If you have any information that you feel belongs in this section, it would be greatly appreciated.

* Windows

Installing Fonts into Windows:

This only covers Windows 3.1. Font is a four-letter word in Windows versions prior to 3.1 due to the distinctions between screen fonts and printer fonts. The upgrade price of Windows 3.1 is justified by the integration of TrueType into the package and the inclusion of useful fonts for all printers.

Commercial fonts usually have installation instructions with their manuals. The approach may differ from the method used for PD and shareware fonts.

To install PD and shareware fonts in Windows 3.1:

1. Copy the fonts onto a suitable scratch area (i.e. a floppy, or any temporary area of your hard disk.
2. Execute "Control Panel" by double-clicking on the icon in the Windows Program Manager's ``main'' group.
3. Double-click on the Fonts icon.
4. Double-click on the "Add" button.
5. Select the scratch directory holding the new fonts.
6. A list of the fonts will be displayed. You can manually select the fonts you like, or you can use the "Select All" button.
7. Make sure the "Copy Fonts to Windows Directory" check-box is checked. This will copy the fonts from the scratch area to your Windows directory.
8. Click on the "Ok" button.

9@Special notes for Windows applications:
Word for Windows (W4W) stores font/printer information in its own initialization files. After you add new fonts, you have to tell W4W that the printer can use the new fonts. Do this by selecting "Printer Setup`` from the W4W main "File" menu item, click on the "Setup'' button, and then click on two ``Ok'' buttons to back out of the setup mode.

Note concerning Windows 3.1 upgrade: There are two upgrade packages available from Microsoft for Win3.1. There is the standard version which contains Type-Type support, and about six font families (Times New Roman, Arial, Courier, Symbols, Wingdings, etc.). It costs something like $ 50 (US). The second version contains a number of TrueType fonts that includes equivalents for the 35 standard Postscript fonts. This adds an additional $ 50, which is a pretty good value. However, if you plan on buying Microsoft's PowerPoint,it includes the same additional fonts/typefaces. So you can save money by not buying the fonts twice.

9@ More about Windows

Why are don't the TrueType fonts that come with Microsoft products (Word-for-Windows, PowerPoint, Windows 3.1 TrueType Font Pack, etc.) display and print properly on my system?

The font matching algorithm in Win3.1 is fairly simplistic. If you install lots of TrueType fonts, the algorithm can get confused. In this case, "lots" is more than 50 or so.

"There is a procedure which Windows must go through when an application requests a font. Each font contains a list of attributes such as Family, FaceName, Height, Width, Orientation, Weight, Pitch, etc. When an application requests a font, it fills out a logical font for Windows containing the necessary attributes, then starts going through a font mapping algorithm to determine

which of the installed fonts most closely matches the requested (logical) font. Penalties are applied against fonts whose attributes do not match the logical font, until the fonts with the fewest penalties are determined. If there is a "tie", Windows may need to rely on the order of the fonts in the WIN.INI file to determine the "winner".
If the fonts you want are in your WIN.INI file, and show up in Windows' Control Panel, then try moving them higher in your WIN.INI file with a file edittor such as SYSEDIT.''

CAUTION: While many Windows 3.1 users would like to have many TrueType fonts at their disposal (and they are many available in the PD) a word of caution. A large number (>50) TT fonts will slow down your windows startup time. This occurs as every installed font is listed in the win.ini file, and Windows has to go through the entire file before starting up. While this may not affect most users, it will especially affect users of Corel-Draw!, so be warned.

* Other Programs

It is an unfortunate fact that almost all MS-DOS programs do things differently. Your best bet is to read the manual that comes with the program you want to use.

3.4. Font utilities

* PS2PK
PS2PK allows you to convert PostScript Type1 fonts into bitmap fonts. The bitmap files produced are in TeX PK format.

* PKtoSFP
PKtoSFP allows you to convert TeX PK fonts into HP LaserJet softfonts.

* PFBDir/PFBInfo
PFBDir and PFBInfo format and display the "headers" in a binary Type1 font.

3.5. Converting Macintosh Type1 fonts to MS-DOS format

Converting Macintosh Type1 fonts into PC Type1 fonts can be done using purely free/shareware tools. I've outlined the procedure below. Make sure you read the "readme" files that accompany many fonts. Some font authors specifically deny permission to do cross-platform conversions.

3.5.1. The tools you need

XBIN	xbin23.zip in /pub/msdos/mac on oak.oakland.edu (or other mirrors)
UNSIT	unsit30.zip in /pub/msdos/mac on oak.oakland.edu
UNCPT	ext-pc.zip in /pub/pc/win3/util on ftp.cica.indiana.edu
REFONT	refont14.zip in /pub/norm/mac-font-tools on ibis.cs.umass.edu
BMAP2AFM	bm2af01.zip in /pub/norm/mac-font-tools on ibis.cs.umass.edu

XBIN converts Mac "BinHex"ed files back into binary format. BinHex is the Mac equivalent of UUencoding, it translates files into ascii characters so that mailers an send them around without difficulty. It also aids in cross platform copying too, I'm sure. BinHexed files generally have filenames of the form "xxx.yyy .HQX".

UNSIT explodes "Stuffit" archives. Stuffit archives generally have filenames of the form "xxx.SIT".

UNCPT explodes "Compactor" archives. The ext-pc implementation is called "extract" and does not require windows (even thought it's in the windows section on cica). Compactor archives generally have filenames of the form "xxx.CPT".

REFONT converts Mac type1 fonts into PC type1 fonts. It also converts Mac TrueType fonts to PC TrueType format. And vice-versa.

BMAP2AFM constructs AFM files from the metric information contained in Mac screen fonts (.bmap files). The screen font files do not have any standard name (although they frequently have the extension .bmap). The screen fonts have file type "FFIL" which, in combination with some common sense, is usually sufficient to identify them.

I've listed the tools that I've used and the sites that are reasonable for me to retrieve them from. It's probably a good idea to check with archie for closer sites if you're not in North America. These tools run under MS-DOS. XBIN and UNSIT can also be run under Unix.

3.5.2. How to do it

Collect the Mac fonts from the archive or BBS of your choice. Most of these files will be in BinHexed format. As a running example, I'm going to use the imaginary font ``Plugh.cpt.hqx''. When I download this font to my PC, I would use the name "PLUGH.CPX". The actual name you use is immaterial.

Run XBIN on PLUGH.CPX. This will produce PLUGH.DAT, PLUGH.INF, and PLUGH.RSR. The data fork of the Mac file (the .DAT file) is the only one of interest to us, you can delete the others.

If the original file had been "Plugh.sit .hqx", we would be using the UNSIT program. Since I chose a .cpt file for this example, I'm going to run UNCPT.

Run UNCPT on PLUGH.DAT. You want to extract the AFM file (if present), the documentation or readme file (if present), and the Type1 outline file. The AFM and README files will be in the data fork of the archive file. The Type1 outline will be in the resource fork. The AFM and README files have Mac "TEXT" type. The Type1 outline file has "LWFN" type. I'm not trying to describe this part in a step-by-step fashion. Use the docs for UNCPT and UNSIT as a guide. If you got this far you probably won't have much difficulty. If you do, drop me a line and I'll try to help.

If the font does not contain an AFM file, extract the screen font. Screen fonts frequently have the extension .bmap and are "FFIL" type files. Use Bmap2AFM to construct an AFM from the screen font. If the archive _does_ contain an AFM file, it's safe to bet that the author's AFM will be better than the one created by Bmap2AFM.

Finally, run REFONT on the Type1 outline that you extracted above. The result should be an appropriate PC type1 outline. REFONT will create a PFM file for you from the AFM file, if you desire.

Remember to register your shareware...

3.5.3. Other comments

* UNCPT is easier to use than UNSIT
* UNCPT has to be run twice. I usually do it like this

```
extract *.cpt -f
extract *.cpt -f -r
```

* When using "unsit30" you probably want the outline file with the MacHeader and the others without it. I think that REFONT requires it but I am not sure.

* REFONT works usually ok. You want a PFA (ASCII) file which is directly usable on NeXT (you may need to convert carriage-returns to newlines but I am not sure if it is necessary).

The biggest problem is with the .afm files that are completely missing or generated by the tools that don't do their job properly.

* BMAP2AFM requires some extra files (ie. other than bmap2afm.exe) to work properly.

3.6. Converting PC Type1 and TrueType fonts to Macintosh format

Refont (version 1.4) can convert (in both directions) between PC and Mac formats of Type1 and TrueType fonts. Note: it _cannot_ convert _between_ formats, only architectures. The procedure described above outlines how to convert a Mac archive into PC format so that you can get at the data. Presumably, the process can be reversed so that ou can get at the data on the Mac side as well. Unfortunately, I don't have a Mac so I can't describe the process in detail.

3.7. Converting PC Type1 fonts into TeX PK bitmap fonts

The release of PS2PK by Piet Tutelaers <rcpt@urc.tue.nl> is a godsend to those of us without PostScript printers. PS2PK converts PC/Unix format Type 1 fonts into TeX PK files. Used in conjunction with the AFM2TFM utility for creating TeX metric files, this allows almost anyone to use Type 1 PostScript fonts. PS2PK is distributed under the GNU License and has been made to run under MS-DOS with DJGPP's free GNU C compiler. The PC version requires a 386 or more powerful processor. Check with Archie for a source near you.

Note: if TeX PK files are not directly usable for you, there seems to be a fair possibility that LaserJet softfonts would be useful. If so, check below for instructions on converting TeX PK files to LaserJet softfonts.

3.8. Converting TeX PK bitmaps into HP LaserJet softfonts (and vice-versa)

There is some possibility that someone will yell 'conflict of interest' here, but I don't think so. I wrote the following utilities:

PKtoSFP	convert TeX PK files to LaserJet (bitmapped) softfonts
SFPtoPK	convert LaserJet (bitmapped) softfonts to TeX PK files

But they are completely free, so I don't gain anything by "advertising" them here. These are MS-DOS platform solutions only. If you know of other solutions, I would be happy to list them.

3.9. TrueType to HP LaserJet bitmap softfonts (HACK!)

If you have the tools, the following suggestion does work, but it isn't easy and it hasn't been automated. To be honest, I haven't really tested it.

If you are using Windows 3.1, get a LaserJet printer driver (you don't need the printer, just the driver). Using the LaserJet driver, direct output to a file and print a simple file containing all the letters you want in the softfont in the font that you are converting. When the print job has completed, the output file will contain, among other things, a LaserJet softfont of the TrueType font you selected. If you know the LaserJet format, you can grab it out of there.

I didn't say it was easy ;-)

This method will not work with ATM [ed: as of 7/92] because ATM does not construct a softfont; it downloads the whole page as graphics.

Here is an overview of the LaserJet bitmap softfont format. It should help you get started. If you have any questions, ask norm <walsh@cs.umass.edu>. If anyone wants to write better instructions... ;-)

Many details are omitted from this description. They are thoroughly discussed in the HP Technical Reference for each model of laser printer. I recommend purchasing the Tech Ref. If you have additional questions and do not plan to purchase the Tech Ref (or do not wish to wait for its arrival), you can ask norm <walsh@cs.umass.edu>.

An HP LaserJet softfont can occur almost anywhere in the output stream destined for the printer. In particular, it does _not_ have to be wholly contiguous within the output file. In fact, fonts can be "intermixed" at will. The following "pieces" make up a font:

A begin font descriptor command (followed by the descriptor) and a series of begin character descriptor commands (followed by their associated data). When a new character descriptor is encountered, it is added to the current font (which may change between descriptors).

In the discussion that follows, the following notational conventions are followed:

Key elements are surrounded by quotation marks. The quotation marks are not part of the element. Spaces within the element are for clarity only, they are not part of the element. All characters (except ESC and # , described below, areliteral and must be entered in the precise case shown).

ESC means the escape character, ASCII character number 27 decimal.

means any decimal number. The meaning of the number is described in the commentary for that element.

* What is a font descriptor?

A font descriptor begins with a font descriptor command and is followed immediately by the data for the descriptor. Font descriptors define data global to the font. In general, more recent printers are less strict about these parameters than older printers.

* What is the font descriptor command?

"ESC) s # W"

In this command, # is the number of bytes in the descriptor. The first element of the descriptor indicates how many of these bytes should be interpreted as the font descriptor (the remaining bytes are commentary only—to the printer, at least). This area is frequently used for copyright information, for example, although some systems insert kerning data into this area.

*What is the font descriptor data?

The data is:

UI	Font descriptor size
UB	Descriptor format
UB	Font type
UI	Reserved (should be 0)
UI	Baseline distance
UI	Cell width
UI	Cell height
UB	Orientation
B	Spacing
UI	Symbol set
UI	Pitch
UI	Height
UI	xHeight
SB	Width Type
UB	Style
SB	Stroke Weight
UB	Typeface LSB
UB	Typeface MSB
UB	Serif Style
SB	Underline distance
UB	Underline height
UI	Text Height
UI	Text Width
UB	Pitch Extended
UB	Height Extended
UI	Cap Height
UI	Reserved (0)
UI	Reserved (0)
A16	Font name
??	Copyright, or any other information

Where

UI =	unsigned integer	SI =	signed integer,
UB =	unsigned byte,	SB =	signed byte,
B =	boolean,	A16=	sixteen bytes of ASCII

After the font name, ?? bytes of extra data may be inserted. These bytes pad the descriptor out to the length specified in the begin font descriptor command.

Note: integers are always in big-endian order (MSB first).

* What is a character descriptor?

A character descriptor describes the character specific info and the layout of the bitmap. Newer printers can accept compressed character bitmaps.

* What is a character descriptor command?

”ESC * c # E”

The # is the length of the descriptor, in bytes.

*What is the character descriptor data?

UB	Format
B	Continuation
UB	Descriptor size
UB	Class
UB	Orientation
SI	Left offset
SI	Top offset
UI	Character width
UI	Character height
SI	Delta X
??	Character (bitmap) data.

Although older printers cannot accept characters that include continuations, newer printers can. If the ”continuation” field is 1, the character bitmap data begins immediately after that byte and the remaining fields _are not_ present.

* Ok, now I understand the data, what do I look for in the output stream?

ESC * c # D defines the font number (remember the number).

ESC) s # W defines the font descriptor (as described above).

ESC * c # E specifies the character code (the #, in this case). The next character descriptor maps to this position in the font. Characters do not have to appear in any particular order.

ESC (s # W defines the character descriptor (as described above).

Remember, these can occur in any order. Experimentation with the particular driver you are using may help you restrict the number of different cases that you have to be prepared for.

3.10. MS-DOS Screen Fonts (EGA/VGA text-mode fonts)

Editors note: the following description was mercilessly stolen from comp.archives on 02SEP92. It was originally Yossi Gil's <yogi@cs.ubc.ca> posting.

FNTCOL12.ZIP contains more than 100 text mode fonts for EGA/VGA displays. It includes fonts in different sizes for Hebrew, Greek, Cyrillic, math symbols and various type styles including smallcaps and script.

It is available at cs.ubc.ca in /pub/pickup/terse/fntcol12.zip

Norman Walsh <walsh@cs.umass.edu>

Monitorn

Kort lektion i PostScript

Postscript är ett programmeringsspråk för att beskriva hur en utskrift skall se ut på en skrivare. Företaget som uppfann Postscript heter Adobe. Språket liknar Forth. Detta gör att språket liksom Forth också arbetar med en stack.

Vill du veta mera om postscript kan du köpa boken "*PostScript Lanuage Tutorial and Cookbook*". Den är utgiven av Addison-Westly och som författare står Adobe Systems Incorporated. Inspiration till artikeln och en del exempel kommer från boken.

True Type är en enklare ersättare för Postscript som Microsoft och Apple har gjort. Med True Type kan man inte göra så avancera saker om med Postscript. Det finns t.ex. inget programmeringsspråk i True Type som gör det möjligt att slumpmässigt välja hur utskriften skall se ut.

Alla exempel här är provutskrivna på HP LasrJet 4 med Postcript och utskrivna på min SUNs bildskärm med Ghostscrit 2.5.2 och pageview. Det är inte så enkelt att skriva ut postscriptkod på en skrivare som är inställd för att automatiskt välja postscriptläge om den får postscriptkod. Det går dock att göra. Jag gjorde det med programvaran psf på Unix.

```
%!
/Helvetica-Bold findfont
40 scalefont
setfont
50 700 moveto
(ABC) show
/Helvetica-Bold findfont
18 scalefont
setfont
142 707 moveto
(BLADET) show
showpage
```

Det här är ett litet postscriptprogram som skriver ut texten ABC BLADET. Alla kommentarer börjar med ett procenttecken och vanligen börjar man ett postscriptprogram med "%!". Ibland brukar det stå "%!PS-Adobe-1.0" eller "%!PS-Adobe-2.0" först. En del program för att skriva ut postscript vill helst att det står %! på första raden så jag börjar med det. 1.0 och 2.0 är versionsnummer för Adobe Postscript.

Vi börjar med att välja en font. Vi tar *Helvetica-Bold* vilket är fet Helvetica. Detta eftersom det är den fonten som används på framsidan av ABC Bladet.

Först lägger vi upp texten *Helvetica-Bold* på stacken och anropar sedan **findfont** operatorn.

```
/Helvetica-Bold findfond
```

findfont letar upp namnet i fontbiblioteket som kallas för **FontDirectory** och placerar detta på ett lämpligt ställe på stacken. Det som vi har fått i retur här innehåller beskrivningar på formen på fonten för en points storlek. För att sätta storleken använder vi operatorn **scalefont**. I programmet skriver vi:

```
40 scalefont
```

vilket sätter 40-point fet Helvetica font. Slutligen gör **setfont** operatorn att fontbiblioteket på stacken blir den aktuella fonten för att användas när vi skriver ut text:

setfont

Origo dvs utgångspunkten för ett postscriptprogram på papperet är det vänstra nedre hörnet. Om vi skriver **0 0 moveto** så placerar vi oss i det nedre vänstra hörnet på papperet. Storleken är uttryckt i 1/72 dels tum. Vi placerar vi oss en bit ner på papperet räknat uppifrån och en bit in:

```
50 700 moveto
```

Den texten som vi skall visa skriver vi genom att skriva:

```
(ABC) show
```

show gör att texten skrivs ut.

Nu vill vi skriva BLADET med lite mindre stil ungefär som det står på ABC-bladets framsida. Därför väljer vi samma font igen, men nu tar vi en lite mindre storlek, dvs 18. Så här ser detta ut:

```
/Helvetica-Bold findfont
18 scalefont
setfont
```

Vi placerar oss lite till höger om ABC och lite ovanför där vi skrev ABC genom kommandot:

```
142 707 moveto
```

Slutligen skriver vi ut texten BLADET genom:

(BLADET) show

För att nu få sidan utskriven så skriver vi operatorn:

```
showpage
```

Observera att detta exempel inte gör det möjligt att skriva ut 8-bitars tecken! Vill vi göra det på en skrivare som har fonter för teckenuppsättningen ISO-8859-1 så kan vi lägga upp en definition som är en slags subrutin. Den kan se ut så här:

```
/ReEncodeISO {
  /ISOLatin1Encoding where {
    pop
    findfont
    dup length dict begin
      {1 index /FID ne {def} {pop pop} ifelse} forall
      /Encoding ISOLatin1Encoding def
      currentdict
    end
    definefont pop
  } { pop pop } ifelse
} bind def
```

För att nu använda en font med ISO teckenuppsättning anropar vi en procedur så här:

```
/Helvetica-Bold-ISO /Helvetica-Bold ReEncodeISO
```

Efter detta så byter vi alltså ut */Helvetica-Bold* i exemplet ovan till */Helvetica-Bold-ISO*. Nu kan vi skriva ÅÄÖ BLADET i stället för ABC BLADET!

```
%!
/Times-Italic findfont 100 scalefont setfont

/printZip
{0 0 moveto (Monitorn) show } def
100 500 translate

.90 -.05 0
{setgray printZip -1 .5 translate} for

1 setgray printZip

showpage
```

Detta är ett lite mer avancerat program. Det skriver ut texten Monitorn med skuggningseffekt. En definition i postscript börjar med ett snedstreck och avslutas med "def". Vi väljer här en riktigt stor kursiv Times font. Möjligen kommer redaktören att minska på detta när han tar in det i tidningen för att få plats, men det är en annan sak.

Siffrorna *.90 -.05 0* placeras på stacken och **for** loopen som står på raden under innebär att loopen går igen från 0.90 till 0 med 0,05 som ökning för varje steg. Det som står mellan { } körs. Det innebär att vi sätter olika gråskalor och sedan flyttar vi oss lite med **translte**.

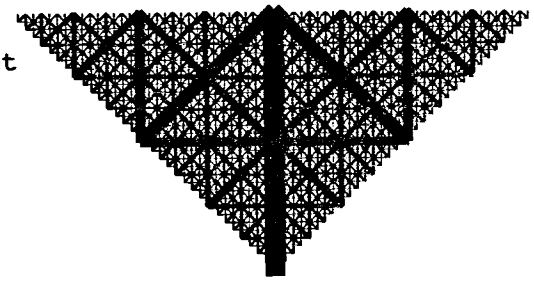
Slutligen sättes gråskalevärdet till vitt och vi skriver ordet Monitorn. Detta sista innebär att bokstäverna blir vita. Utan detta hade bokstäverna blivet helt fyllda.

```
%!
/depth 0 def
/maxdepth 10 def
/down {/depth depth 1 add def} def
/up {/depth depth 1 sub def} def

/DoLine                    % Skriv vertikal linje
 { 0 144 rlineto currentpoint
   stroke translate 0 0 moveto} def

/FractArrow
 { gsave .7 .7 scale       % Reducera skalan
   10 setlinewidth         % Sätt linejbredd
   down DoLine             % Skriv rad
   depth maxdepth le
    { 135 rotate FractArrow
     -270 rotate FractArrow} if
   up grestore } def

% Här börjar programmet
200 500 moveto
FractArrow
stroke
showpage
```



Detta är ett exempel på ett avancerat program som ritar en triangel med en rekursiv rutin. I stället för show används här stroke för att skriva ut grafiken. le betyder mindre än eller lika med och tillhör if operatorn som står lite längre ner.

Här är ett trevligt program som skriver ut en text som definieras i början efter /note. Vad programmet gör är att det skriver ut allt med olika fonter och typer så att det ser ut som ett utpressningsbrev som är klippt från en tidning.

<1789!
Bo Kullmar

```
%!PS-Adobe
% Ransom note writer.  Version 1.00
% (C) Copyright 1992 Diomidis Spinellis.
% All rights reserved. You can use and copy this
% program as long as the notice above remains %
intact.Modified copies must be clearly
% marked as such. Send comments, threats,
% corrections etc. to dds@doc.ic.ac.uk
% Put here the ransom note you want
% Use the | symbol to generate new lines.
/note (Mr. Diomidis Spinellis|
This is a message from an international terrorist
group. We have abducted your files and have hidden
them away. You must pay $500 by Sunday afternoon
in order to see them again.
You will receive further telephone instructions.
Do not notify the police or any authorities; you
will endager their life.| Remember: time flies!
) def

% Left margin position
/left 30 def
% Right margin position
/right 510 def
% Line advance
/lineskip -40 def

% Print the string
% string ransom -
/ransom {
      {chr ransomchar}
      forall
} def

% Convert a character code to a string
% int chr string
/chr {
      ( ) dup 0 3 index put exch pop
} def

% Draw a black filled box for the bounding box of
% the character char box -
/box {
      gsave
            gsave
                  true charpath pathbbox
            grestore
            1 add /ury exch def
            1 add /urx exch def
            1 sub /lly exch def
            1 sub /llx exch def
            newpath
            llx lly moveto
            urx lly lineto
            urx ury lineto
            llx ury lineto
            closepath
            0 setgray
            fill
      grestore
} def

% Display a single ransom character
% character ransomchar -
/ransomchar {
      dup (|) eq {
            pop
            crlf
      } {
            randfont 20 random 30 add scalefont
1>**                                  setfont
            dup
            gsave
                  5 random rotate
                  3 random 1 sub 3 random 1 sub
1>**                                  rmoveto
                  dup ( ) eq {
                        [ {show} ]
                  } {
                  [
                        { show }
                        { show }
                        { true charpath stroke }
                        { true charpath 10 random
1>**                    20 div setgray fill }
                        { dup box 1 setgray
1>**                                  show}
                  ]
                  } ifelse
                  randarray exec
            grestore
            stringwidth
            pop 0 rmoveto
            format
      } ifelse
} def

% Return a random value 0 <= r < int
% int random int
/random {
      rand
      exch
      mod
} def

% Return a random element from the array
% array randarray any
/randarray {
      dup
      length
      random
      get
} def

% Return a random font
% - randfont font
/randfont {
      [
            /Times-Roman
            /Times-Bold
            /Times-Italic
            /Times-BoldItalic
            /Helvetica
            /Helvetica-Bold
            /Helevetica-Oblique
            /Helevetica-BoldOblique
            /Times-Roman
            /Times-Bold
            /Times-Italic
            /Times-BoldItalic
            /Helvetica
            /Helvetica-Bold
            /Helevetica-Oblique
            /Helevetica-BoldOblique
            /Courier
            /Courier-Bold
            /Courier-Oblique
            /Courier-BoldOblique
      ]
      randarray
      findfont
} def

% Do a carriage return / line feed
% - crlf -
/crlf {
      currentpoint exch pop
      left exch moveto 0 lineskip rmoveto
} def

% Move to a new line if we are at the end of it
% - format -
/format {
      currentpoint
      pop
      right gt
      {crlf}
      if
} def

clear
erasepage
initmatrix
left 700 moveto
.5 setlinewidth
note ransom showpage
 $dd
```

Anmärkning
Vi har tvingats att markera på några ställen där orginalet har endast en myclket lång rad och vi har föredragit att bryta raden. Markeringen har gjorts med tecknen 1>**.

Här bredvid visar vi resultatet av Bosses exempel och en förminskning av "utpressarbrevet"
Redaktionen

ABC BLADET

Något om rutiner i BASIC-II

Supersnabb input/output av data i BASIC-II

En av de starka sidorna hos BASIC-II är t ex förekomsten av VAROOT, en funktion som ger minnesadressen till den s k variabelroten. För en strängvariabel innehåller variabelroten dimensionerad längd (två bytes), adress till variabelns data (två bytes, samma som VARPTR), samt aktuell längd (två bytes, samma som LEN). Dessa värden kan avläsas och ändras med PEEK och POKE. I BASIC-II manualen fanns inga direkta exempel på användningen, bara en varning om att den är till för "avancerad programmering". Så småningom lärde sig folk ändå att använda VAROOT, till stor del tack vare "Bit för bit med ABC800". En stor fördel med att ha tillgång till variabelroten är bl a att man i programkoden kan kontrollera dimensionerad längd och aktuell längd, även inne i en funktion, om man skickar VAROOT av en strängvariabel som parameter. Det behövs annars komplicerade språk som C++ för att konstruera strängklasser med motsvarande möjligheter. Microsofts olika BASIC saknar helt de möjligheter som VAROOT-funktionen ger.

VAROOT kan bl a användas till ungefär samma sak som pekare i andra språk som C och Pascal.
Här är ett exempel:

```
10  A$="Detta är sträng A$"
20  Root=VAROOT(A$)
30  POKE VAROOT(B$), PEEK(Root), PEEK
    (Root+1), PEEK(Root+2), PEEK(Root+3), PEEK
    (Root+4), PEEK(Root+5)
40  ! A$ och B$ står nu för samma data
50  ; A$
60  ; B$
```

Det går också att definiera olika variabeltyper "ovanpå varandra". Motsvarande en union i C. Man kan då uppnå samma sak som vid funktionen CVT, men för en hel vektor åtgången. Det kan i sin tur användas för att väsentligt snabba upp dataöverföring till/från en fil.

För praktiskt bruk undviker man nog att använda PRINT och INPUT(LINE) för datahantering till/från fil. GET och PUT är mycket kraftfullare. För att spara säg 10000 heltal på fil kan man göra ungefär så här:

```
10 DIM Tal(10000)
.
.
(Vektorn Tal() får sina värden)
.
.
100 PREPARE "TAL.DAT" AS FILE 1
110 FOR I=1 TO 10000
120 PUT #1 CVT%$(Tal(I))
130 NEXT I
140 CLOSE 1
.
```

(Inläsning kan ske på motsvarande sätt med CVT$%)

Nackdelen med metoden är att BASIC-tolken använder sin mesta tid till att snurra runt i FOR-loopen. För att snabba upp det hela skrev jag som novis på BASIC-II program-mering en assemblerrutin för ABC800. Det tog sin tid, för jag var nybörjare även på Z80 programmering, och vad som stod i BIT FÖR BIT angående DOS-anrop i ABC800 var långt ifrån glasklart. Efteråt kom jag på ett mycket vettigare sätt att lösa problemet, nämligen att använda VAROOT och lägga en strängvariabel "över" vektorn med tal. Det blev kortare, mycket enklare och bara någon hundradels sekund långsammare än assemblerkoden. Så här exempelvis:

```
10   ! Exempel på Snabb I/O av data från fil i BASICII
20   !
30   Max=10000 ! - högst ca 32000
40   DIM Tal(Max),Tal$=0
50   Ptr=VARPTR(Tal(1))
60   POKE VAROOT(Tal$),2*Max,
     -SWAP%(2*Max),Ptr,SWAP%(Ptr),2*Max,
     SWAP%(2*Max)
70   ! Strängen Tal$ svarar nu mot heltalsvektorn
     Tal(). (Varje heltal= 2 bytes)
80   !
90   ! Lägg lite värden i vektorn
100  FOR I=1 TO Max : Tal(I)=2*I : NEXT I
110  !
120  ! Så här skriver man. Det går snabbt.
130  ; "Skrivning, Start: " TIME$
140  PREPARE "DATAFIL" AS FILE 1
150  PUT #1 Tal$
160  CLOSE 1
170  ; "Klart: " TIME$
180  ;
190  ; "Läsning med LOOP, Start: " TIME$
200  OPEN "DATAFIL" AS FILE 1
210  FOR I=1 TO Max : GET #1 T$ COUNT 2 :
     Tal(I)=CVT$%(T$) : NEXT I
220  CLOSE 1
230  ; "Klart: " TIME$
240  ;
250  ; "Läsning utan LOOP, Start: " TIME$
260  OPEN "DATAFIL" AS FILE 1
270  GET #1 Tal$ COUNT 2*Max
280  CLOSE 1
290  ; "Klart: " TIME$
```

Mitt ursprungliga problem gällde att snabba upp inläsning av data från en indexfil. Kanske finns det fortfarande någon som använder BASIC II/PC och som kan dra nytta av tekniken.

Om parameteröverföring i BASIC-II, eller självförbättrande kod?

Parameteröverföringen i BASICII är i viss mån "smart" åtminstone vad gäller strängar. När en funktion anropas med en strängparameter, så överförs VAROOT för strängen, och BASICII-tolken använder det överförda värdet för att få access till parametervärdena. Om parametern är ett stränguttryck så utvärderas parametern och den utvärderade strängen läggs på stacken. VAROOT för det uttryck som finns på stacken överförs som parameter. För parametrar som bara består av en strängkonstant eller strängvariabel skapas ingen kopia på stacken, utan VAROOT för konstanten överförs direkt. Detta är ju smart så tillvida som det bl.a. sparar utrymme under själva körningen.

Det är naturligtvis ganska meningslöst att inne i en funktion tilldela ett värde till en överförd parameter under sådana omständigheter. Är parametern ett sammansatt uttryck, så måste en tilldelning vara helt missriktad. En strängkonstant skall naturligtvis inte ändras under exekveringen. Möjligen skulle man vilja tilldela värden till en enkel strängvariabel som överförs, och på så sätt få vad som i Pascal motsvaras av VAR-deklaration. Inne i funktionen är det ändå omöjligt att särskilja vid den kontroll som sker i det s k fix-up skedet. I konsekvens av detta vägrar BASICII att godta vanliga tilldelningar till funktionsparametrar inne i en funktion.

Min vän Mats Lindgren gjorde mig en gång uppmärksam på att denna spärr i BASICII i viss mån kan kringgås när det gäller strängar, genom att använda MID$ i vänsterledaet av tilldelningen. Jag brukar kalla det Lindgren-effekten. Man kan anse det vara en brist eller en bug i BASIC-II om man så vill. Observera att stränglängden aldrig kan ändras på detta sätt. När det gäller uttryck är det i allmänhet poänglöst, eftersom uttrycket på stacken glöms bort efter genomlöpt funktion, så man kan aldrig se resultatet utanför funktionen. Gäller det strängkonstanter, kan man ändå uppnå resultat som verkar förbluffande. Om man inte har ovanstående betraktelse i minnet vill säga.

Knappa in nedanstående lilla program! Gör sedan gärna LIST på det och kontrollera att allting kommit in rätt.

```
10 DEF FNUtskrift(A$)
20 MID$(A$,1,15)='Ett bra program'
30 PRINT A$
40 RETURN 0
50 FNEND
60 !
70 Z=FNUtskrift('Ett dåligt hack')
```

Vad utskriften vid körning kommer att bli, är inte svårt att lista ut, men kör det i alla fall! Lista efter körningen programkoden igen, och begrunda programlistan!

Självreproducerande program

Att skriva program som skriver program, en "programgenerator", behöver inte vara så avancerat som det kanske låter. Ett litet exempel var t ex programmet "POSTGÖR", som publicerades i ABC-bladet för ett antal år sedan, och som alstrade programsnuttar i BASIC-II för att skapa poster lite i stil med RECORD-typer i pascal eller struct i C. En intressant övningsuppgift är att i valfritt programspråk skriva ett program (så kort som möjligt) som har sin egen källkod som output. När man kör programmet skall det alltså skriva ut en exakt kopia av sin egen källkod. BASIC-II/PC är kanske något passé som programspråk, men ett självreproducerande program i BASIC-II blir ganska kort och lättfattligt.

Frågan om självreproducerande program togs bl.a. upp i Dr Dobbs Journal #180 (sept. -91) av Ray Valdés i artikeln "Littel Languages - Big Questions". I någon mån som ett test på hur "kraftfullt" ett språk är. Ett speciellt pris utlovades till den som först klarar att skriva ett självreproducerande program i Lotus Macro Language. Den relativa lättheten att skriva ett självreproducerande program i BASIC-II antyder att det skulle vara ett någorlunda kraftfullt språk. I andra språk blir det lätt väldigt kryptiskta program.

Det ger kanske inte så mycket att bara se och provköra ett sånt här program. Visst är det förbluffande att få samma resultat på bildskärmen vid RUN som vid LIST, men vad som ger en intellektuell stimulans är att konstruera ett sådant här program på egen hand. Alltså: försök att (i ditt favoritspråk) konstruera ett program som genererar sig självt som output. Det ger en viss "kick" och insikt i programmering om man klarar av det. Ofta blir det en slags aha-upplevelse.

För den som inte har tid, ork eller tror sig om att klara det på egen hand, ges här en programlistning på en BASIC-II variant. Om inte annat kan det fungera som ledtråd för den som vill pröva samma sak i PASCAL, C, Forth, Assembler eller något annat. C-varianten blir avsevärt längre, och att få till det i Pascal är kanske ganska knepigt? Den som behärskar Lisp skall kunna göra ett mycket kort program. Exempel i C, Lisp och i språket Mumps fanns i ovan nämnda nummer av DDJ.

```
10 DATA "FOR I=10 TO 80 STEP 10"
20 DATA" READ A$ : IF I=40 RESTORE
   10"
30 DATA" ; NUM$(I); : IF I<50 ; ' DATA
   ' CHR$(34) A$ CHR$(34) ELSE ; ' '
   A$"
40 DATA "NEXT I"
50 FOR I=10 TO 80 STEP 10
60 READ A$ : IF I=40 RESTORE 10
70 ; NUM$(I); : IF I<50 ; ' DATA '
   CHR$(34) A$ CHR$(34) ELSE ; '' A$
80 NEXT I
```

Jag har också funderat på att i BASIC-II/PC kan man kanske gå via dos och pula in ett LIST-kommando i tangentbordsbufferten, så att BASIC-tolken efter avslutat program listar detsamma, men det är fusk!

<7804>
Jan Sundström

"You are not expected to understand this"

Vi har fått följande lilla snuttt insänd av <2397>Johan Persson

"You are not expected to understand this" Detta är en berömd kommentar som återfinns i källkoden till kärnan i UNIX. Jag bad en gång Dennis Ritchie att förklara det, och här är hans svar.

This is a famous comment (in fact I have two sweatshirts on which it is written; UKUUG had them made).

It wasn't quite so callous as it may seem. It was in the kernel, in the part of the system that has the basic coroutine, process-switching mechanism. I've appended the nearby code from the 6th edition system. savu() saves the state of the process; aretu restores it, including switching stacks and moving to the new process. The amusing thing is that these routines are flawed; they can't be implemented properly in all machine architectures. In other words, we didn't "understand this" either at the time. In the 7th edition system (ca. 1977) the details changed significantly.

Dennis

```
/*
* Switch to stack of the new process and set
* up his segmentation registers.
*/
retu(rp->p_addr);
sureg();
/*
* If the new process paused because it was
* swapped out, set the stack level to the last
* call to savu(u_ssav). This means that the
* return which is executed immediately
* after the call to aretu actually returns from
* the last routine which did the savu.
*
* You are not expected to understand this.
*/
if(rp->p_flag&SSWAP) ä
        rp->p_flag &= üSSWAP;
        aretu(u.u_ssav);
å
/*
* The value returned here has many subtle
* implications.
* See the newproc comments.
*/ return(1);
```

Min elektroniska lekstuga

En datorphreaques bekännelser

För 25 år sedan träffade jag min polare Kai för första gången. Då byggde vi radiosändare och kämpade tillsammans för att klarade tekniska provet i radioteknik och telegrafiprovet. När jag fåttmitt A-certifikat och satt där och dah-dih-dah-dit-tade på min hypermoderna transceiver, kunde jag ju icke i min vildaste fantasiföreställa mig vilka skojiga prylar vi skulle få att leka med 25 år senare -- i transistorernas och de integrerade kretsarnas värld, för att nu inte tala om de nuvarande 386:or och 486:or med en hisklig kapacitet jämfört med den allra första datorn jag någonsin kom i kontakt med. Den tingesten upptog ett jättestort rum och kunde ha använts som interiör i rymdskeppet Enterprise i TV-serien Startrek.

Vi är varken ingenjörer eller tekniker jag och min kompis, utan bara elektroniska 'phreaques'.

Långt före persondatorernas tid började vi leka med RTTY (radio teletype) dvs sådant som Lasse Strömberg <7872> skrev om i nummer 1 av ABC-bladet 1993. I något surplus-lager hittade vi televerkets gamla Creed 7b 5-baud-teletype, 1927 års modell. Vi var igång med en svindlande skrivhastighet som låg en bit under 10 tecken per sekund. Vidundret slamrade och levde om, så XYL som vi radioamatörer kallar tigern i huset krävde att den skulle kastas ut per omgående. Men nu hade det dykt upp något slags 'glasburkar' där bokstäverna syntes på en TV- skärm, således bildskärmsterminal. Så kom de första IMSAI-8080-datorerna i byggsats. För den fattige fanns inget annat val än att löda ihop en dylik.

Jag arbetar som dokumentöversättare och visste sedan min gymnasietid att det fanns skrivautomater med hålremsor, så att man kunde slippa skriva samma saker om och om igen. Att få äga en sådan drömde jag om. Det visade sig dock att IMSAI gick att använda till primitiv ordbehandling.

Sedan gick det raskt framåt med datortekniken och man hann knappt med. Sinclair. ABC80. ABC800. Jet80. Commodore etc. etc. Det snackades om att IBM var på gång med någon ny burk med ett häftigt operativsystem.

Datorer och elektronik är i dag så billiga ting att vi verkligen kunnat komma loss med vår elektroniska lekstuga.

Om grindar

I hopp om att inspirera andra teknikstollar till någon rolig eller praktisk lösning i det egna hemmet, tänkte jag helt kort nämna en del grejer som jag implementerat i min lilla översättarlya, där det finns en gallergrind, en s.k. säkerhetsgrind.

När grinden är stängd påverkas ett relä av en mekanisk kontakt, så att alla telefonapparater är bortkopplade från min lilla kontorsväxel. I stället går linjerna till varsin telefonsvarare. På så sätt störs inte grannarna av en massa ringsignaler när jag inte är där. Jag behöver heller aldrig tänka på att slå på eller av telefonsvararna. När grinden öppnas är telefonsvararna fortfarande aktiva i två minuter så att jag skall hinna ta av mig ytterkläderna och ev vittja telefonsvararna på inkomna samtal, och så att inte ett pågående samtal till telefonsvararen skall brytas i onödan. Sedan kopplas automatiskt alla linjer bort från telefonsvararna och styrs nu till växelns anknytningar. Kontoret är öppet. Vill jag ta emot samtal tidigare nollställer jag tidskretsen meden knapp. Vill jag inte ta några telefonsamtal startar jag en tidskrets ställd på 20 minuter (bra när jag har t ex har besök). Vill jag lägga mig och sussa på en soffa någon halvtimme, så skjuter jag till gallergrinden och slipper bli störd av ringande telefoner.

Priviligierad pocket och töntiga tekniker

En sann elektronikgalning har givetvis en Telepocket bärbar telefon från televerket. Även om alla telefoner är avstängda, så kan jag ringa ut på den, eftersom en optokopplare känner av att 'husse' vill ringa.

Ett vanligt irritationsmoment för mig är att tekniker som tagit fram alla möjliga apparater inte begripit att de också skall gå att stänga av. Televerkets Telepocket har en ringsignal som inte går att stänga av. Den ihärdige kan dock öppna hela basenheten och slita loss kontakten till den högtalare som används för att kvillra fram det obehagliga ljud som torde kunna väcka döda. Ett problem är att om man lämnar in den på service, så tror televerkets reparatörer att det är något 'vajsing'och sätter i kontakten igen före leverans.

Min faxapparat gick inte heller att stänga av. Alltså, skruva i sär fanskapet, döda ringsignalen.

Teknisk dans eller dansant teknik?

Jag är inte enbart teknikgalen, utan också fanatisk hobbydansare (stildansare), dansar slowfox, wienervals, cha-cha, rumba etc och brukar ofta träna på olika steg och turer när jag tar en paus i jobbet. Om någon ringer på telefonen så dämpas stereohögtalarna automatiskt och går ned på mycket låg nivå tills samtalet är slut. På så sätt slipper jag gå och skruva ned ljudet själv.

I mitt arbete som översättare använder jag en s k VOXBOX för syntetiskt tal, samma burk som blinda brukar koppla till sin dator för att få höra text uppläst elektroniskt.

Jag var inte nöjd med standardprogrammet, så jag fick hitta på en massa grejer som passar mina behov som t ex att hoppa över visst slag av text typ streckade rader, punktade rader och en massa annat. Det är ju det som är tjusningen med datorer och programmering, att man äntligen kan få skräddarsy lösningar helt efter sina egna behov.

En egen BBS som står och puttrar i ett hörn är ett måste och jag har möjlighet att ringa in till denna BBS och sända in ett vanligt BBS-brev som kan omvandlas till fax-brev och sändas ut med fax-modem (programmet kommer från Thomas Lundgren på Träffpunkt 44).

Praktisk AXE med fel i!

Tack vare AXE-systemet finns det också en massa praktiska tjänster direkt från televerket så att man slipper uppfinna så mycket själv. Vidarekoppling vid upptaget är en 'höjdare' som jag verkligen rekommenderar för den som har två eller flera telefoner. Om det är upptaget på linje 1 så kopplas samtalet till linje 2 och vice versa. Det verkligt skojiga var att vi inte fick denna tjänst *68# att fungera, och att televerket på fullaste allvar trodde att det var s k SBS-fel ('skit bakom spakarna'). Efter många om och men visade det sig att någon hade programmerat AXE-enheten fel uppe på Kungsholms telestation. Nu fungerar det som det skall.

Pillat på kranarna

När det gäller min dator och den ordbehandling jag kör i Wordperfect, så har jag också varit inne och pillat en hel del på olika kranar och har en Wordperfect som har alla Wordstars kommandon och en massa hiskeliga specialmakron som lämpar sig för mitt jobb.

Detta innebär dock att jag inte så lätt kan byta till andra program, t ex Windows och Word för Windows. Ofta förklarar vänner och bekanta att jag är hopplöst omodern som fortfarande kör renodlad DOS, men det skulle bli för många timmar av grottande i grejerna för att byta upp sig (ned sig?).

Nere

Just nu ligger uppfinnandet lite nere för mitt vidkommande, och det är bara att konstatera. Datorutvecklingen har gått så snabbt och burkarna blivit så billiga att de nu faktiskt är gemene mans egendom. Imponationsfaktorn har sjunkit. Jag måste antagligen sända upp en rymdfärja från kontoret för att imponera på vänner och bekanta.

Men jag kan ju glädja mig åt att jag har grejer som underlättar jobbet och gör det trivsammare.

Bill Leksén

Linux och Linus Torvalds

Jag lovade att jag skulle rapportera lite grand om det föredrag som Linus Torvalds höll i Linköping i samband med Lysator:s 20-årsjubileum.

Mest för att likna System V Unix

Linus Torvalds studerar matematik och datateknik på Universitet i Helsingfors. Han visst inget om vad UNIX var när han började utbildningen men fick se mer och mer varefter han införskaffade ett exemplar av Minix men tyckte att den inte kunde tillräckligt. Detta gjorde att an började skriva Linux även om den inte hette det från början. Den skrevs då mest för att likna en System V UNIX men det har ändrats så idag så skrivs den mest för att stämma överens med POSIX-standarden.

Arrogans, ignorans, envishet och inspiration

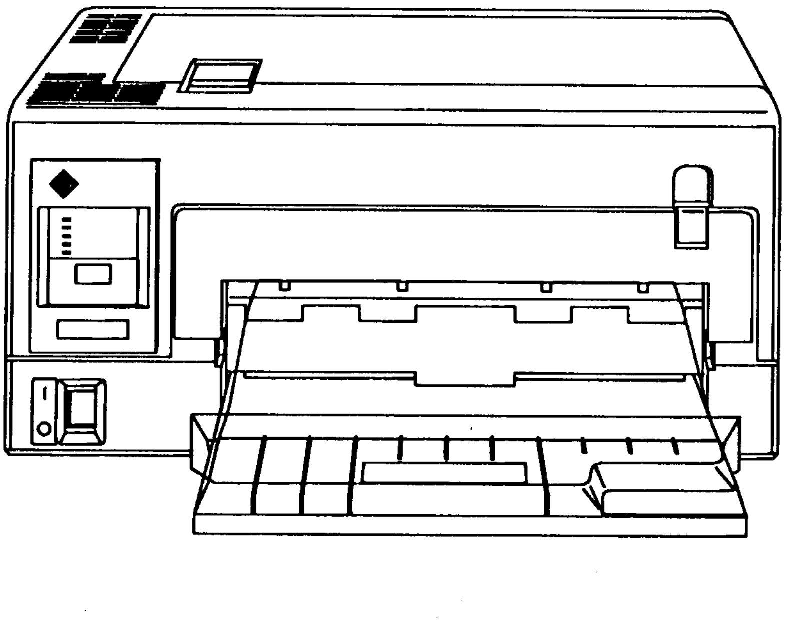
Linus menade att enda möjligheten med att klara ett projekt som att skriva Linux var att ha arrogans, ignorans, envishet och inspiration. "Man ska helst inte veta om att man inte klarar av ett projekt för att klara av det" sa Linus.

Som vi alla vet så har Linux blivit ganska populärt. Det finns idag uppskattningsvis 10.000 personer som använder Linux över hela världen och det är ca 40.000 personer som läser newsmötet comp.os.linux. Det är två doktorsavhandlingar på gång som använder sig av Linux. Däremot så finns det ingen organisation alls bakom Linux. Det är enligt Linus en av de stora anledningarna till att utvecklingen sker så snabbt som möjligt. Det är ju många som har haft med utvecklingen av Linux att göra även om alla inblandade inte har programmerat själva. Ett exempel är en tysk som ville köra Linux och GCC med 2Mb minne. Det var det som fick Linus att implementera virtuellt minne så snabbt som han gjorde. Det blir ju så att om programkoden är fri så blir det snabb betatestning eftersom alla får tillgång till allting.

Versionsnummer

En sak som många frågar sig är naturligtvis när version 1.0 ska komma men det tar nog ett bra tag enligt Linus. Han hoppade i versionsnumreringen direkt från 0.13 till 0.95 och det har byggt upp lite för optimistiska förhoppningar. Den version som är den nyaste idag (1993-03-28) är 0.99.7A som jag kommer att ladda ner i programbanken. Lysator har installerat den versionen och den fungerade bra av vad jag kunde se även om man var tvungen att kompilera om ps för att den skulle fungera.

<8054>
Jan Smith



Föreningsbrev B

Anmäl adressändring i tid

ABC klubben
Box 14 143
161 14 Bromma

Begränsad eftersändning
Vid definitiv eftersändning återsändes försändelsen med nya adressen på baksidan

Medlemspriser T-märkta höghastighetsmodem

	Medlemspris	exkl moms
Hidem 14400DLX	**5 900:-**	4 720:-
Hidem 14400DLX internt	**5 400:-**	4 320:-
V.32bis (14400 bps), MNP5, och V.42bis , sv handbok, 2 års garanti.		
Hidem 14400FAX	**6 900:-**	5 520:-
Hidem 14400FAX Pocket	**6 900:-**	5 520:-
Hidem 14400FAX internt	**6 400:-**	5 120:-
Samma som ovan plus faxfunktion, sändning och mottagning upp till 14400 bps.		
Hidem 9600DLX	**4 900:-**	3 920:-
Hidem 9600DLX internt	**4 400:-**	3 520:-
V.32 (9600 bps), V.42, MNP5, och V.42bis , sv handbok, 2 års garanti.		
Hidem 2442FAX	**2 700:-**	2 160:-
Hidem 2442FAX Pocket	**2 490:-**	1 992:-
Hidem 2442FAX internt	**2 100:-**	1 680:-
2400 bps, gränssnittshastighet upp till 9600, MNP5 och V.42bis , fax sändning och mottagning i 9600 bps, sv handbok, 2 års garanti.		
U.S. Robotics Sportster, Fax	**4 500:-**	3 600:-
V.32bis (14 400 bps), MNP5, V.42bis, Grupp III fax (9600 bps).		
U.S. Robotics V.32bis, Fax	**7 000:-**	5 600:-
V.32bis (14 400 bps), MNP5, V.42bis, Grupp III fax (9600 bps), 1 års garanti.		
Hidem 2400P, Pocketmodem med sändfaxfunktion	**1 600:-**	1 280:-
V.22bis (2400 bps), sändning av fax i 4800 bps. Batteri och batterieliminator ingår.		
ExtraLine, linjedelare utvecklad i Sverige	**1 600:-**	1 280:-
Låt en telefon, ett modem och en telefax samsas på en telefonlinje.		

Till faxmodemen medföljer DOS-faxprogram (BITFAX resp BLASTFAX).

Samtliga modem uppfyller kraven i lagen om teleterminalutrustning (SFS 1992:1527) och får anslutas till det allmänt tillgängliga telenätet. Reservation för prisändringar. Vid förskottsbetalning ingen fraktavgift, annars tillkommer frakt- och PF-avgift.

NHN-Systemkonsult , Nils Hansson <519> 08-669 80 20 Fax 08-658 69 33
Wollmar Yxkullsgatan 33, 118 50 STOCKHOLM 08-560 355 37 Pg 433 34 85-3