

STACK POINTER

2004-1. Organ för Datorföreningen STACKEN, KTH.



Datorföreningen STACKEN



TACKEN är en kårförening inom Tekniska Högskolans Studentkår. Föreningen bildades 1978. Dess ändamål är att ge teknologer och anställda vid KTH möjlighet att friare utveckla sitt datorintresse.

Vi sysslar med det mesta, t.ex. underhåll av STACKENS datorsystem, programmering på alla nivåer, datornätverk, m.m. I STACKENS regi kan medlemmarna med större frihet utveckla de intressen, som skolans ordinarie datorresurser inte ger utlopp för. Föreningen har idag ca 250 medlemmar, huvudsakligen teknologer på KTH. På torsdagskvällar brukar vi träffas

i föreningslokalen på Drottning Kristinas väg 47A.

Medlemssavgiften 2004 är 104 kronor för studerande och 204 kronor för icke studerande. Medlemsansökan görs enklast på <http://www.stacken.kth.se/>

Datorföreningen STACKEN
% NADA
KTH
100 44 STOCKHOLM

E-post: styrelsen@stacken.kth.se
Telefon: 08-791 87 97
Postgiro: 433 01 15-9



STACKPOINTER



STACKPOINTER är organ för Datorföreningen STACKEN på KTH. Den utkommer med ett varierande antal nummer, i den takt bidrag inkommer.

Det har gått många år sedan det senast kom ut en STACKPOINTER, så många att de flesta av STACKENS nuvarande medlemmar bara har hört talas om att föreningen en gång i tiden haft en tidning. Vi i redaktionen hoppas att STACKPOINTER i fortsättningen skall kunna komma ut mer regelbundet. Vi vill därför passa på att tacka de som skrivit artiklar och samtidigt uppmana alla er som inte gjort det, att skriva ar-

tiklar till nästa STACKPOINTER. Man behöver inte vara superexpert på det man vill skriva om, utan det räcker att man är någorlunda insatt och har viljan att skriva något. Om du är intresserad av att skriva för STACKPOINTER eller har artikelförslag eller något annat du vill vädra, får du gärna höra av dig till redaktionen.

E-post: stackpointer@stacken.kth.se
 I redaktionen: Jan Michael Rynning och
 Harald Barth
 Foto: Harald Barth
 Färdigställd: 2004-10-21

Innehåll

Introduktion till AFS	4
STACKENS datorsystem	8
Ettor och nollor på spåret	10

Rivningsbilder	19
H l k r t	20
God of backup	21

På gång



UNDER hösten har STACKEN varit med på kårens dag, hållit öppet hus och ordnat en serie intressanta föreläsningar. Just nu håller vi på med programmeringstävlingar. Det är tänkt att det skall bli en per vecka, men det är inte frågan om att vara "bäst", utan mer om att hitta på roliga och originella lösning-

ar med högt "hackvärde". Torsdagen den 20:e november är det höstmöte. Vid årsskiftet går hyresavtalet för den nuvarande föreningslokalen ut, så vi blir tvungna att flytta igen. Exakt hur det blir med ny lokal och flytt är inte känt än, men en sak är säker: Det kommer att behövas medlemmar som hjälper till!

Introduktion till AFS

av Hans Insulander, bearbetad av Harald Barth



FS är ett distribuerat filsystem till Unix och är gjort för stora nätverk och en växande datormiljö. AFS använder en klient/servermodell, där filer lagras på filservrar. Klienten hämtar den data som behövs, och mellanlagrar informationen lokalt på klientdatorn. AFS tillhandahåller även ett mycket flexibelt system för att sätta rättigheterna på filer i systemet, som gör det möjligt för användaren att dela filer med sina kollegor på ett bra sätt. AFS är också betydligt säkrare än NFS, som vanligen används i Unix.

Tidigare tillverkades och supportades AFS av IBM. Numera finns en fri version som kallas OpenAFS¹. IBM har vid det här laget lagt ner supporten för den kommersiella AFS-versionen.

I STACKENS regi utvecklas även en oberoende fri klient som kallas ARLA².

Varför AFS?

- Lättare att arbeta i grupper

Varje användare kan dela ut rättigheter på sina filer till vissa specifika personer eller en grupp personer. Varje användare kan skapa egna grupper.

- Bättre driftssäkerhet

Om en servermaskin av någon anledning inte fungerar slutar inte hela systemet att fungera — endast de filer som finns på den maskin som slutade fun-

gera kommer vara otillgängliga. De filer som fanns på en maskin som eventuellt inte fungerar kan lättare återställas, antingen genom att man flyttar själva hårddisken eller genom att man återskapar filerna från bandbackup.

- Replikering

Filer som inte ändras ofta, t.ex. applikationer som installerats, kan replikeras ut till flera maskiner. Replikering innebär att samma filer finns tillgängliga på flera olika filservrar. AFS ser automatiskt till att hålla filerna uppdaterade, och delar belastningen mellan olika servermaskiner.

- Lättare administrera filsystemet

AFS är från början gjort för att vara enkelt att administrera i en växande datormiljö. T.ex. kan man flytta filer mellan hårddiskar och datorer utan att den som använder filerna märker det. Man kan till och med flytta filerna under tiden filerna används utan att det märks. Börjar en hårddisk bli full kopplar man bara in en ny (eventuellt en helt ny filserver) och flyttar över en del kataloger till den nya hårddisken.

- Distribuerat filsystem

Filerna i AFS finns tillgängliga på alla AFS-kapabla datorer i hela världen, förutsatt att de kan prata via lokalt nätverk eller via Internet. Det innebär att

¹<http://www.openafs.org/>

²<http://www.stacken.kth.se/projekt/arla/>

du kan låna en dator på något annat ställe och använda dina filer precis som vanligt. Om du har konto på flera olika institutioner inom KTH kan du använda filerna oavsett var du befinner dig, förutsatt att institutionerna har en AFS-klient installerad.

- **Tillgänglighet till backuper**

Skulle du råka radera en fil finns en kopia från gårdagen tillgänglig under katalogen `.OldFiles` i din hemkatalog. Backupkopian är en icke skrivbar kopia av dina filer och skapas varje natt klockan 00:00.



Begrepp i AFS

Cell

AFS är indelat i s.k. celler. En cell är en administrativ enhet, dvs. alla datorer och filer som tillhör cellen administreras som en grupp. T.ex. gäller användar- och gruppidentiteter för cellen. Om du har "utökade" rättigheter av något slag gäller de inom din cell. Namnet på cellen är helt valfritt, men vanligen använder man domännamnet. STACKENS afs-cell kallas följaktligen `stacken.kth.se`.

Katalogen /afs

När man installerat AFS på sin dator får man en ny katalog på datorn. Katalogen heter `/afs` och ofta kallas den katalogen för AFS rot-katalog. I `/afs`-katalogen finns underkatalogen `/afs/stacken.kth.se` som är STACKENS cell. Övriga AFS-celler som finns med i den officiella listan finns under `/afs/<cellens_namn>`.

Volym

Ett delträdd av filstrukturen lagras i något som heter volym och ser ut som en virtuell hårddisk. Varje användare brukar ha en egen hemkatalog som ligger i den egna hemvolymen och som även har en individuell filsystemsquota.

ACL:er

Till varje katalog i AFS finns en s.k. ACL kopplad. ACL är en förkortning för Access Control List, en behörighetslista. ACL:en listar vad olika användare har rätt att göra med dina filer, t.ex. läsa, skriva, radera osv. T.ex. kan du göra så att dina labkamrater kan skriva i en viss katalog i din hemkatalog.

Grupper

I ACL:er går det även att använda grupper, t.ex. kan man göra så att alla användare i gruppen teachers kan läsa dina labbrapporter.

Token

För att använda filer i AFS måste du ha en s.k. token. Enkelt förklarat är en token en krypterad sträng som bekräftar din identitet. Din token lagras i minnet i datorn och och skapas när du loggar in. Den är också tidsbegränsad för att försvåra för någon att stjäla din identitet. Om du inte har en token eller om den gått över tidsbegränsningen kan du inte använda dina filer. Sin token får man oftast i samband med sin Kerberos ticket.

Realm

En Realm är för Kerberos vad en cell är för AFS. Men eftersom autentifiering till AFS sker oftast med hjälp av Kerberos så använder man realmen för att autentificera sig. I motsats till cellen och domänen brukar den vara döpt till samma namn fast med stora bokstäver vilket för STACKEN blir STACKEN.KTH.SE. Om man har kompicerade organisationsstrukturer eller vill röra till det så kan man ha flera celler tillhörande en realm, men oftast hör en realm till en cell och en domän.

Filrättigheter i AFS

ACL — Access Control list

ACL:en listar användare och vad de har rättighet att göra, t.ex. kan man sätta rättigheterna så att användarna Kalle och Ada kan läsa och skriva i katalogen telekomprojekt_2, medan användaren Nisse

bara kan läsa.

ACL:er kan endast sättas per katalog. De filer som finns i katalogerna använder katalogens ACL. Nyskapade kataloger "ärver" ACL:en från den katalog den skapas i.

ACL:erna administreras helt och hållet av användarna själva. Du kan även bestämma att en annan användare får administrera dina ACL:er.

Grupper i AFS

Användaren ola kan skapa gruppen ola:telekom och sedan dela ut rättigheter på katalogen telekomprojekt_3 till den gruppen. Läger ola sedan in nya personer i gruppen uppdateras ACL:en automatiskt på alla de kataloger där gruppen används. Grupper som skapas av enskilda användare har alltid namn enligt formatet <username>:<groupname>, där <username> är namnet på personen som äger gruppen. Administratören kan skapa så kallade prefixlösa grupper, dvs. grupper utan användarnamn och kolon.

Om man har kompisar så kan man skapa en egen grupp som innehåller dessa. Som kuriosa kan nämnas att det även finns motsatsen, dvs. om man har fiender så kan man skapa en grupp med dessa och sedan ge den gruppen negativa rättigheter i en ACL vilket betyder att alla utom medlemmerna i gruppen får göra filoperationerna.

Skillnader mellan AFS och ett lokalt Unix-filsystem

Rättigheter

Delar av rättigheterna som används i det lokala filsystemet är inte relevanta i AFS. Man kan sätta rättigheterna för group och

other men de är inte giltiga i AFS, trots att de visas precis som förut. Det gäller alltså att man inte litar på den information som visas när man listar filerna med `ls`-kommandot i Unix. Istället måste man använda AFS-specifika kommandon för att sätta och lista rättigheterna i AFS. Dessutom så gäller rättigheterna på katalogbasis, det går inte att ha en enstaka fil med annorlunda rättigheter i en katalog. Sådana problem går dock oftast att lösa med mjuka länkar.

Token

Token behövs för att läsa filer, om du inte har en token kan du inte läsa dina filer. En token skapas automatiskt när du loggar in men har bara en begränsad livslängd på

några timmar. Så när giltigheten har gått ut måste man förlänga den genom att logga ut och in igen eller enklare genom att använda kommandot `kauth`.

Struktur

Antalet filer under `/afs` är stort. Mycket stort. Program som försöker söka igenom alla filer under `/afs` brukar inte bli klara. Katalogerna under `/afs` bildar oftast men inte alltid ett träd, dvs. det går att konstruera cirkulära vägar som program som `find` kan fastna i.

Övrigt

Det går inte att ha hårda länkar mellan filer i olika kataloger. En fil skrivs först tillbaka till servern när den blir stängd.

STACKENS AFS-cell bygger på redundant och tillförlitlig hårdvara.



STACKENS datorsystem

av Hans Insulander, bearbetad av Harald Barth



STACKENS datorsystem består till allra största delen av fem till tio år gamla Sun-arbetsstationer, som skänkts till föreningen av främst KTH, Sun och Ericsson. Även en del PC och Mac finns representerade, samt enstaka maskiner från Digital. I stort sett alla maskiner är skänkta till STACKEN, men några privatägda datorer finns också lämnade till STACKENS förfogande.

Mjukvarumässigt kör maskinparken till en övervägande del Solaris, men även NetBSD, OpenBSD, FreeBSD, Linux i olika varianter, VMS och Windows finns representerade i olika omfattning.

Förutsättningar och mål

Solaris-installationen har byggts med antagandet att maskinerna har minst två gigabyte hårddisk och minst 128 megabyte RAM. NADA och ITE har skänkt oss en del Sun Ultra-maskiner, och alla Sunnar är uppgraderade till sådana.

Grundtankarna med installationen är:

- Enkelt att installera och ominstallera maskiner
- Automatiskt underhåll av alla maskiner samtidigt
- Installation av programvara ska vara så enkelt som möjligt
- Man ska kunna ha flera versioner av samma program installerade samtidigt
- Driftssäkert — dvs. systemet ska inte gå sönder p.g.a. någon liten detalj

- Lätt att sätta sig in i — dvs. man ska lätt kunna förstå hur det fungerar

Underhåll av arbetsstationer

När man installerar en ny maskin, eller installerar om en redan befintlig, behöver man bara sätta igång det automatiska installationsprogrammet som Sun skickar med Solaris. Någon kvart senare är maskinen uppe och man kan logga in på den som en vanlig användare, och köra alla sina program precis som vanligt.

Det är viktigt att undvika att installationen "ruttnar" — dvs. att maskinerna med tiden blir olika. För att undvika detta satsar vi på att allt underhåll ska ske via shellskript o.dyl. som ser till att maskinerna är rätt konfigurerade, så normalt sett görs inga ändringar direkt på maskinerna.

Installation av programvara

Vi håller den mesta programvaran installerad i AFS, eftersom detta gör det mycket enklare och snabbare att installera och underhålla den. Vissa kritiska programvaror som Kerberos, AFS-klienten m.m. kopieras dock automatiskt till den lokala hårddisken.

För att kunna ha flera olika versioner av ett visst program installerade samtidigt valde vi att installera alla program i katalogen `/mpkg/<program>/<version>`. Anledningen till detta är att man ofta inte kan vara säker på att en nyare version

av programmet är ”bättre” än den gamla. T.ex. kan den nya ha okända buggar, eller så kanske någon viss användare faktiskt föredrar den gamla versionen.

För att det inte ska bli så jobbigt att ställa in alla sökvägar och annat som behövs för att köra ett visst program, finns ett slags hjälpprogram vid namn ”modules”. Till modules skriver man små styrfiler som beskriver hur skalet ska ställas in för att kunna köra ett visst programpaket.

Om man av någon anledning vill köra en äldre version av ett program kan man ange att man vill ladda in modulen för den äldre versionen. Anger man ingen version får man den som systemadministratören har ställt in som default.

Servrar

De flesta servrar är Sunnar som är väldigt lika arbetsstationerna. Det som skiljer att de startar en annan uppsättning program vid boot. Undantaget är AFS-servrarna och Kerberos-servrarna som är specialinstallerade för att just vara sådana. Eftersom det är så få, så har vi valt att göra detta för hand istället för att automatisera. En annan sak som skiljer två av AFS-servrarna från alla andra är att de är köpta. Detta för att vi inte hade passande datorer när vi flyttade till mindre lokaler.

Hur kan du hjälpa till?

Vad som görs med STACKENS datorsystem är helt beroende på vad medlemmarna själva gör — ingenting kommer att hända automatiskt. Det man vill se gjort, får man göra själv.

Därför ser vi gärna att de som är intresserade av att hjälpa till med da-

tordriften hör av sig, lämpligast till staff@stacken.kth.se, så kan vi hjälpa varandra att få en ännu bättre datormiljö. Det är inte nödvändigt att vara ”expert” inom sitt område; det räcker med att man har ett intresse och vill lära sig mer.

Just nu saknar vi personer som är intresserade av dessa områden:

Databas Vi skulle behöva en intresserad att underhålla t.ex. en MySQL- och/eller en PostgreSQL-databas.

KDE och GNOME Det skulle vara trevligt om nån vill ta hand om en eller båda av dessa.

Webb Det är så mycket som kan göras med en webbserver idag, så här finns det alltid att göra.

Email Det vore trevligt att få en modernare setup.

Om du vill hjälpa till med detta eller något annat, får du gärna höra av dej eller helt enkelt dyka upp i lokalen några torsdagar.

Framtiden

Konkreta planer för framtiden inkluderar:

- Uppgradering till Solaris 10.
- Utveckling av arbetsstationerna. Tanken är att man ska kunna välja mellan Solaris och ett annat OS (BSD eller kanske Linux) på våra arbetsstationer — såsom vi hade för ett par år sedan, men den här gången ska vi försöka göra det möjligt att underhålla maskinerna.
- Arla och OpenAFS — göra en underhållen installation på arbetsstationerna. Förmodligen kommer vissa maskiner köra Arla och vissa OpenAFS.

Ettor och nollor på spåret

av Harald Barth



RÅN att ha varit använd av några få, så börjar nu digitaltekniken göra intåg på spåret på bred front. Till och med traditionellt inriktade klubbar som SMJ³ börjar införa digitaldrift. Vad vill då modelljärnvägaren uppnå med detta? Som alltid så är vi ute efter så "bra" tågtrafik som möjligt. Bra är ett mycket subjektivt begrepp, men följande nämns ofta: Oberoende styrning av alla lok. Oberoende styrning av alla växlar. Minimering av antalet kablar under modelljärnvägen. Handkontrollrar som antingen är löstagbara eller till och med trådlösa. De flesta håller med om att tekniken skall vara till för att kunna köra små tåg och inte tvärtom.

Loken på spåret

Spåret begränsar på ett naturligt sätt antalet ledare mellan omvärlden och loket till två, eller med fungerande luftledning tre ledare. Inte ens så kallade *Märklinister* använder fler ledare. I forntiden har det dock funnits försök till multipelstyrning med fler ledare (t.ex. Trix Express) och fler frekvenser (t.ex. Zimo). Begränsningen till två anslutningar är både bra och dålig. Å ena sidan får man mycket färre kablar, å andra sidan så måste information och kraft överlagras. Idag görs överlagringen m.h.a.

digital teknik. I loket, växeln eller ljussignalen sitter sedan en liten krets som avkodar informationen från drivströmmen. Informationen är kodad enligt ett digitalt protokoll. Detta kallas Digital Command Control på engelska och förkortas DCC. Den digitala kodningen görs idag främst på två sätt: Märklin/Motorola och NMRA DCC⁴. (NMRA är det amerikanska standardorganet för modelljärnväg, NEM heter den europeiska motsvarigheten. NEM verkar först nyligen ha upptäckt DCC). Märklins protokoll har vid det här laget reverse engineerats och NMRAs protokoll har från början haft öppen specifikation. Den finns att hämta på NMRAs sida om standarder⁵. De flesta tillverkare håller på att gå över till att stödja styrprotokollet enligt NMRA. Sedan finns det några mindre system på marknaden med egna standarder, men dessa kommer antagligen att själv dö så småningom. Trix med sitt alldeles egna system som kallas Selectrix är lite av tågsystemens Betamax (någon som kommer ihåg Betamax video?) om man jämför med originalsyste met från Märklin/Motorola. Den största skillnaden för en *likströmsrallare* jämfört med tidigare är att loket innehåller styrningen och att körriktningarna på körkontrollern därför är framåt respektive bakåt för loket istället

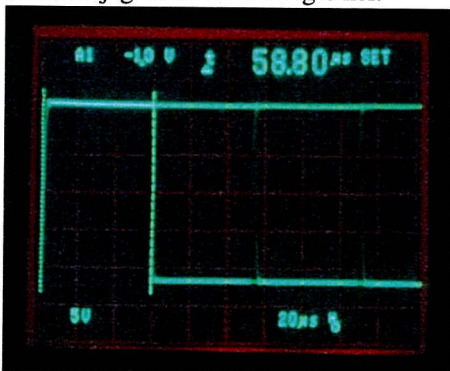
³<http://www.smj.org/>

⁴<http://www.nmra.org/>

⁵<http://www.nmra.org/standards/>

för åt ena respektive andra hållet på spåret. Även om det inte är lika tydligt som hos ett ånglok, så har även elektriska och diesellok en fram- och en bakände, ofta utmärkt på något sätt. Riktningstyrning i loket har funnits i Märklinlok med hjälp av ett omkopplingsrelä sedan urminnes tider.

Det måste dock poängteras att trots tillverkarnas tappra försök att binda oss modellrallare till sitt system, så är val av digitalsystem oberoende av val av räls. Trots att de flesta som kör med trerälsystemet använder Märklin/Motorola och alla med tvärräls använder NMRA DCC så finns det inget som föreskriver detta. Det går till och med att blanda dessa system, hur kommer jag att beskriva längre ner.



Men hur ser nu den digitala signalen ut elektriskt på spåret? Alla digitalsystem är uppbyggda kring en växelströmssignal i form av en rektangelvåg. Jag kommer här att exemplifiera med NMRA DCC-signalen eftersom dess specifikationer finns lätt tillgängliga för alla, men andra digitalsystem fungerar på liknande sätt. En NMRA-signal på spåret består av en växelströmssignal som är symmetrisk runt

noll så att likströmslok som råkar stå på samma spår inte rör på sig.

Rektangelvåg som utgör signalen (den kallas så eftersom den ser ut så om man tittar på den med hjälp av ett mätinstrument, ett oscilloskop) varierar mellan två spänningsnivåer, $-12V$ och $+12V$. Beroende på tågskalan kan spänningen också vara lite högre. Informationen som skall överföras till loket representeras av spänningsnivåernas växlingar mellan dessa två nivåer. En växling mellan dessa två spänningar med signalen $52\mu s$ (mikrosekunder) på $-12V$ och $52\mu s$ på $+12V$ tyds som en etta. Detta kan skrivas som J . På motsvarande sätt representeras en nolla av en växling med signalerna minst $100\mu s$ på $-12V$ och minst $100\mu s$ på $+12V$. Detta kan skrivas som J eller J . Om ni nu blir förvirrade när jag förklarar hur DCC-signalen ser ut elektriskt så håll ut. De elektriska detaljerna är i och för sig kul att känna till, men inte nödvändiga för att använda digitalsystemet.

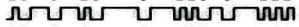
Exempel:

Binärt:	1 0 0 1 0 0 1 1 0 0 1 1 1
Vågform:	
Bitnr.:	12 3 45 6 789 A B C D osv.

Om man tittar på värdet ovan, så kan lista ut att det är MSB (mest signifikanta biten) först som gäller. Alla håriga detaljer finns i själva NMRA-standard 9.2⁶.

Eftersom man nu också vill att det skall gå att styra ett gammalt analogt lok i någon slags "bakåt-kompatibilitetsläge", så använder man sig av en teknik som kallas "zero stretching". Det betyder att man gör nollorna asymmetriska och därmed får en likströmskomponent:

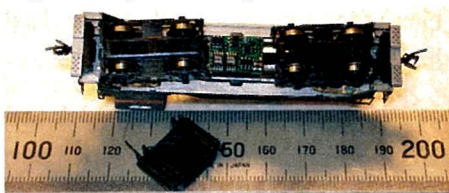
⁶http://www.dcc.info/standards_rps/S-92-2004-07.pdf

Binärt:	1 0	0	1 0	0	1 1 0	0	1 1 1
Vågform:							
Bitnr.:	12	3	45	6	789	A	BCD osv.

Oberoende hur mycket nollorna förlängs (upp till ett maxvärde förstås), för digitala lok är detta samma kommando som ovan. I exemplet ovan är nollorna förlängda olika mycket. Genom det kan likströmskomponenten anpassas till en viss spänning.

Detta gör att man kan köra ett likströmslok i den strömkretsen. Man kommer inte riktigt upp i $\pm 12V$ som förr, men då alla lok i alla fall är överväxlade det är inte något problem. Vi modellrallare är tyvärr medvetna om att det har uppmätts hastigheter på omräknat 500km/h på stackars ånglok. Zero stretching får man dock till priset att loket tjuter med den valda frekvensen och att man eldar bort mera värme som möjligtvis skadar motorn. Jag har inte råkat ut för det själv men hört talas om att motorer har tagit skada. I riskzonen ligger framför allt N-skalelok (1:160) med små 3-poliga motorer. Så det blir inte riktigt tummen upp för den lösningen. I längden orkar man i alla fall inte lyssna på tjutet på ungefär 9kHz och kommer att utrusta alla sina modeller med dekoder.

Ingående systemkomponenter



Dekoder

Dekoder kallas det lilla chippet som avkodar signalerna. För varianten som klarar av

en ström på en ampere har vi idag kommit ner till storleken eller snarare litenheten på $4 \times 9 \times 3,8\text{mm}$ och ett pris på cirka 250kr, så det är inte längre ekonomiskt att bygga själv. Dessutom används det ytmonterade specialkomponenter. Riktigt så litet behöver det dock inte vara om man inte som jag har sina lok i mindre skala än majoriteten. Dekodrar finns i alla smaker där de enklaste endast kan styra hastighet och ett lyse medan de mer avancerade varianterna klarar fler specialfunktioner och lastreglering genom att känna av strömspikarna när kollektorn roterar. EMF heter det visst. Vad man kan använda fler specialfunktioner än lyse till begränsas av platsen i loket, strömförbrukningen och fantasin. Följande ha setts (eller hörts): Blinkande varningslampa (typiskt USA), hel/halvljus (typiskt Sverige), horn eller vissla, motor eller ångljud (så klart växlande med hastigheten), rök, varningspip och slammer vid dörrstängning, musik i dansvagn samt höj- och sänkbar arbetsplattform (vilket är överkurs). Det går också rykten om en modellrallar som nästan helt slutade köra tåg när han hade börjat använda sin fjärrstyda kranvagn.

Dekodrar finns i stort sett av fyra olika typer:

1. Lokdekoder för NMRA-protokollet
2. Lokdekoder för Märklinprotokollet
3. Tillbehörsdekoder för NMRA-protokollet
4. Tillbehörsdekoder för Märklinprotokollet

Bland dessa finns sedan undertyper som bland annat skiljer sig genom hur många adresser de klarar av. De flesta dekoder lyssnar endast på en typ av protokoll, även om en del dekoder av märkena

Centralenhet	Protokoll			
	Motorola/Märklin	DCC	FMZ	Selectrix
Märklin 6020	X			
Digitrax Chief		X		
Roco		X		
Lenz		X		
Uhlenbrock Intellibox	X	X		X (1)
Fleischmann Twincenter		X	X	X (2)
Arnold	X	X		
Zimo	X	X		
DDLs erdded	X	X		
TrackOne	X	X		

(1) Inte samtidigt
(2) Icke dokumenterad

Arnold och ESU⁷ lyssnar på datapaket av såväl NMRA- och Märklinformat, dvs. är en korsning av 1 och 2.

Centralenhet

Digitalsignalen genereras av något som man kallar centralenhet. Den innehåller ett antal kretsar som lägger ut körspänningen med styrinformationen om och om igen. Centralenheter har från början också varit för endast ett system, men på senare år har det kommit centralenheter som klarar av att generera fler än ett protokoll. Centralenheter kan vara inbyggda system eller ett program som körs på en dator.

Tabellen innehåller endast de största tillverkarna/formaten och de som jag personligen har kommit i kontakt med. Här ser man också att vissa protokoll är mer utbredda än andra och framtiden kan vara oviss för alla utom de två stora.

Inbyggda system

Centralenheter finns att köpa som inbyggda system, helt enkelt en liten låda med µCPU som kan en enda sak.

Märklin 6020 och 6021 är en typiska sådana centralenheter. De är dessutom inget annat, inte ens en nödstoppsknapp har man kostat på sig. Det var lite snålt.

Det finns centralenheter som kan lite mer och även inkluderar andra funktioner som körkontroller eller datorinterface: Lenz har en som heter Compact och Uhlenbrock en som heter Intellibox. Lenz är känd för att konstruera och tillverka åt Atlas, Märklin och Arnold, Uhlenbrock åt Fleischmann.

På tågföreningen har vi bestämt oss för en Digitrax centralenhet, enda problemet med den hittills har varit den undermåliga bruksanvisningen.

Dator med program

En PC med ett speciellt program kan ersätta en inbyggd centralenhet med sin lilla CPU som kör ett enda program vilket startas när man slår på lådan. Mer om detta nedan.

Körkontroller

Märklin 6035 är en lokkontroller av klas-

⁷<http://www.loksound.de/>

sätter ihop de olika systemkomponenterna sida vid sida. Uhlenbrock och Arnold har också passande komponenter till PC-bussen. Lenz kallar sin bus XBus och Digitrax kallar sin LocoNet. Av alla dessa är LocoNet det mest intressanta eftersom det är en riktig databuss med CSMA/CD à la Ethernet, dock genom att man sänker en ledare med 50mA mot jord. Gränsspänningen är 4V. Digitrax har publicerat specifikationen för hemmabruk.

LocoNet använder en 6-ledare (RJ12) där pinnarna 2+5 är jord och 3+4 är LocoNets databärare i två parallella ledare med samma potential. Sedan brukar man förvirra användarna genom att man lägger ut en strömbegränsad kopia av rälssignalen mellan ledare 1 och 6 i samma kontakt. Den kan användas för att ansluta fler boosters samt används för att spänningsmata handkontrollarna, men inte för att överföra information till handkontrollarna.

Interface

För att koppla sin dator som kontrollenhet till centralenheten finns det olika lösningar från tillverkarna. Märklin har sitt interface 6050 som tar in kommandon via RS232 och skickar dessa vidare till den interna kommandobussen.

För LocoNet finns en färdig, dyr och dålig (ingen galvanisk avskiljning) samt en billigare och bättre hårdvara i byggsats så att datorn kan bli en LocoNet-enhet och därmed LocoNet-kontrollenhet. Rent teoretiskt skulle datorn även kunna vara LocoNet-centralenhet, men jag har inte sett detta i praktiken än. Det är kanske lite programmering kvar för att förverkliga detta.

Uhlenbrocks Intellibox har redan inbyggt RS232. Så om du vill styra centralenheten från datorn, glöm inte att räkna med kostnaden för interfacet om det inte redan är inbyggt.

Lenz har ett interface kallat LI100 mellan XBus och RS232.

Booster och rälssignal

Om man har en stor anläggning så vill man dela upp den i ett antal strömmatningsområden. Detta för att begränsa kortslutningseffekten som man annars skulle kunna få. Det är fult med svetsmärken på räls, lokens hjul och strömvattningsarbleck, för att inte prata om hur fult det är med sönderbrända plastdetaljer. Ja, det har hänt. Även elsäkerhetsfolket skulle ha synpunkter på en matning på 12V och 500VA. En kortslutningsström på 41A kan vara farlig även med endast 12V. Istället för att bygga ett jätteeffektsteg så distribuerar man en lågeffektssignal till ett antal förstärkare som var och en är effektbegränsade till säg 5A. Naturligtvis lite mera till trädgårdsbanan i skala 1:22,5 och lite mindre till hyllbanan i skala 1:160.

För dem som inte är hemma i skalor: Den vanligaste skalan som alla tänker på när det gäller modelljärnväg heter H0 (uttalas halv-noll eftersom den är hälften så stor som den som heter noll) och motsvarar 1:87.

Sedan har jag inte sagt att det inte finns personer som töjer lite på reglerna om strömstyrkor på sin trädgårdsbana eftersom det inte är någon vits med ett flertågssystem om det inte finns tillräckligt effektuttag för körning av flera lok.

Centralenhet	I ² C	XBus/XpressNet	Kommandobuss		
			LocoNet	MausBus	SRCP + TCP/IP
Märklin 6020	X				
Digitrax Chief			X		
Roco		?		X	
Lenz		X			
Uhlenbrock Intellibox	X		X	X	
Fleischmann Twincenter	X		X	X	
Arnold	X	X			
DDLs erddcd					X
TrackOne					X

Vändslingeautomatik

Du som kör Märklin med mittkontakt kan hoppa över detta avsnitt.

Alla banor med tvärälssystem, oberoende om de körs med digital eller analog drift har ett vändslingeproblem när de två motsatta räler med olika polaritet träffar varandra i en vändslinga eller diagonal. I digitaldriftsfallet är problemet dock mycket lätt att lösa: Man isolerar vändslingan eller diagonalen och kopplar in en liten dosa som vänder polerna ifall det blir kortslutning. Eftersom färdriktningen som bekant sitter i loket, så fortsätter tåget åt samma håll i alla fall. Se bara till att vändslingespåret är så långt att hela tåget får plats, anars blir det kortslutning genom tåget som överbryggas de isolerande rälsskarvarna. Vändslingeautomatiken kan ibland vara inbyggd i effektförstärkaren, ibland behöver man köpa till i en liten låda.

Transformator

All denna elektronik vill bli matad med ström. Som brukligt i MJ-sammanhang använder man 16V växelström. Om du inte vill köpa de dyra "digital"-transformatorerna som tillverkarna säljer och inte heller har kvar en gammal trotjänare från

analogtiden, kan du köpa likvärdigt någon annanstans. Du behöver titta på spänning (14–16V), ström (2–6A beroende på behov) och effekt. Effekt är spänning gånger ström, dvs en transformator som levererar 4A vid 16V har en effekt av 64VA. Transformatorer märks i VA och inte i W (watt), men i vårt fall kan vi säga att det är samma sak. Transformatorn skall dessutom vara säkrad även på lågspänningsidan och det kan vara bra med en automatsäkring. Jag avråder alla från hemmabyggen eftersom en felaktig koppling på transformatorn kan leda till döden. Det finns fina kapslade transformatorer att köpa som är värda sina pengar.

Dator (t.ex. PC)

Det mesta som man kan göra med ett inbyggt system kan man även göra med lämplig programvara. Dock har en PC vanligtvis inte någon utgång som kan leverera godtycklig vågform (inte färdigtänkt sidotanke: ljudkortet då?) Eftersom man just vill slippa att köpa extra hårdvara och många PCr har en serieport, så kan man generera den kontinuerliga vågformen som man behöver till DCC med lite genial programmering samt udda parametrar (custom divisor till UART) på se-

rieporten. Efter detta behöver serieports-signalen endast förstärkas med en booster. Jag har hört talas om köpprogramvaror som heter WinLok och så finns det DDL som står för Digital Direkt för Linux som jag har goda erfarenheter av. DDL kan dock ställa till med småproblem för den oinvidde eftersom sådana detaljer som dokumentation och mailinglistor domineras av tyskan. Nyligen har det även dykt upp en "port" till Win* som heter DDW.

Der Mobas digitalprojekt⁸, där hittar man länkar till DDL, DDW och annat av samma sort.

DDL

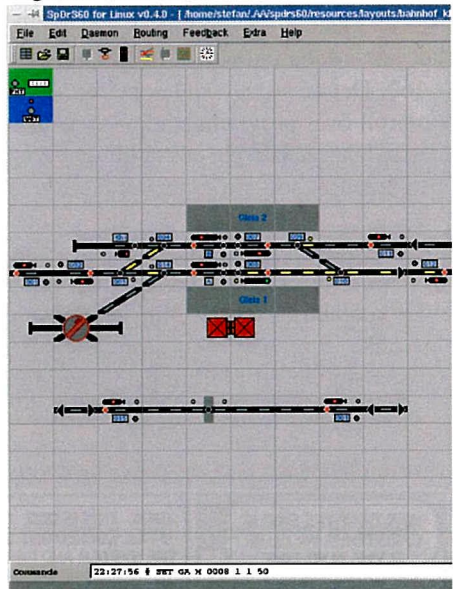
DDL implementerar ett klient/server-system där diverse kontrollprocesser som har hand om ett hastighetsreglage eller ett virtuell ställverkspulpet kommunicerar m.h.a. SRCP (Simple Railroad Control Protocol) till en demon som sedan styr själva tågbanan. SRCP är ett textprotokoll över TCP precis som SMTP dock med kommandon för att sätta hastigheter på lok och lägga om växlar. Det finns ett antal klienter som pratar SRCP. Jag känner till följande:

- Gamepad som lokkontroll
- Mus och fönster som lokkontroll
- Eget skriptspråk för till exempel automatiserad pendling
- Enkla textkommandon (för debuggning)
- Ställverk

SRCP och LocoNet

En LocoNet devicedriver för Linux är un-

der utvecklig och kopplingen till SRCP diskuteras. Detta är dock inte riktigt klart än. Med detta skulle det även finnas möjlighet att använda datorn som LocoNet-centralenhet eller koppla datorbaserade SRCP-klienter som t.ex. SpDrS60 till en Digitrax-centralenhet.



”SpDr60 för Linux”⁹ är en simulation av ett SpDrS60-ställverk¹⁰.

Att köra

Skillnaden när man kör tåg ensam är inte så stor, dock om man börjar köra ett större antal personer tillsammans upplever man en stor frihet i att slippa bry sig hur saker och ting är kopplade under banan. Man pluggar helt enkelt in sin handkontroller i ett hål på banans framkant, väljer loknummer och kör. Detaljer som att inte två

⁸<http://www.der-moba.de/Digital>

⁹<http://www.linux-modellbahn.de/>

¹⁰http://www.stellwerke.de/formen/seite2_m.html

personer skall använda samma lok samtidigt håller centralenheten koll på, och för det mesta funkar det utan problem. Dock har det hänt att Digitrax centralenheten har "glömt bort" en association mellan lok och handkontroller och då vägrar ge mig det loket på en timeout av 200 sekunder. Det känns lite dumt, speciellt om loket just i detta ögonblick råkar susa fram i 160. Det är ett av skälen att jag vill ha kontroll över programvaran i centralenheten. Men så hör jag också till den gruppen personer som programmerar själv.

Återkoppling

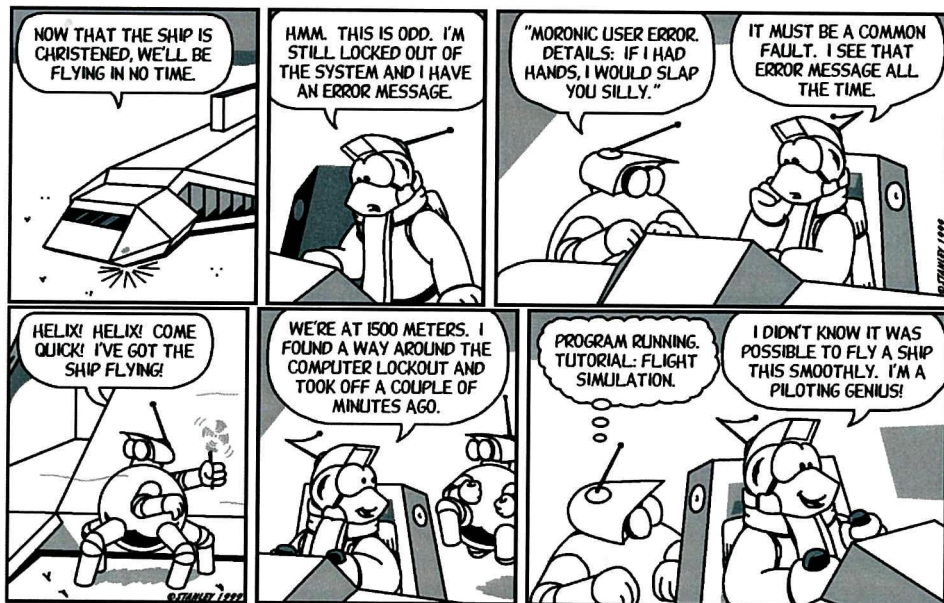
För att överföra information från växlar eller kontakträls (spårledningar) finns det lite olika lösningar. Mest utbredd är S88-bussen som är ett Märklinpåhitt och fun-

gerar som ett stort skiftregister. Andra alternativ är Lenz XBus eller LocoNet som även kan användas till sådant.

Framtiden

Det som håller på att hända nu är att som komplement till återkoppling av växellägen och upptagetillstånd på spår, så kommer det direkt återkoppling från loket (främst hastighet), samt igenkänning av individuella tåg m.h.a. data från dekodern för att exakt kunna veta var tågen befinner sig. Om detta innebär en ökning av körglädjen eller bara blir till nöje för automatiseringsfreakarna får framtiden utvisa. Tillverkarna hoppas i alla fall redan nu kunna presentera nya funktioner innan de ens hunnit lösa de leveransproblem som ofta uppstår på elektronikfronten.

Freefall by Mark Stanley, published with kind permission



Rivningsbilder

av Harald Barth



Här är ytterligare några bilder från rivningen av STACKENS gamla lokal för ett år sedan.



Hålkort

av Thomas Nyström



N gång i tiden fanns det något som hette 'hålkort'. Ett hålkort var en kartongbit med hål i och där en dator kunde både läsa och skriva (eller stansa) sådana kort. Det fanns också speciella maskiner som bara kunde stansa hålkort, sådana maskiner använde man i stället för bildskärmar. Dessa hålkort kunde då användas för att mata in information till datorer. Nedanstående historia tilldrager sig på den tiden då hålkorten fortfarande fanns kvar men det var nästan ingen som använde dylika kartongbitar utan de flesta hade gått över till bildskärmar.

Det hände sig en dag på den stora datacentralen QZ (Stockholms datamaskincentral för högre utbildning och forskning) att en av dess programmerare satt och hackade på en av datacentralens datorer hemifrån (modem fanns faktiskt på den här tiden). Han får då reda på (via KOM, originalet alltså) att det finns ett fel i ett av hans program. Han bestämmer sig därför att han 'skall bara' fixa detta fel i programmet. Han fixar felet och kontrollerar att programmet fungerar. Han har nyligen börjat lära sig UNIX på egen hand (detta var när UNIX började sitt intåg i den riktiga världen) och när han vid ett tillfälle skall kopiera programmet så råkar han ge UNIX-kommandot 'cp' istället för kommandot 'copy' som det heter på det här datasystemet (Tops-10).

Vårn programmerare blir lite förvå-

nad när han får ett kryptiskt meddelande i stil med:

`%BATINF - CPUNCH job started`

Han inser plötsligt att han skrev fel kommando och ger därefter rätt kommando så att programmet kopieras dit han vill.

Nästa morgon när våran programmerare inträder på sin arbetsplats så börjar han naturligt med att passera sitt postfack. I postfacket hittar han en radskrivarlista från datacentralens Amdahl-system. Nedanför postfacken ser han också ett antal kartonger med hålkort och han undrar i sitt stilla sinne vem som fortfarande använder hålkort.

Amdahl var en tillverkare av datorer som gjorde stordatorer som var kompatibla med IBM-stordatorer och detta tilldrar sig ju på en riktig datacentral så man måste ju ha en riktig IBM-stordator. I alla fall nästan. Han läser igenom listan och finner innehållet mycket komiskt där han kan se att han har kört ett batch-jobb på Amdahl-systemet och kan då dra sig till minnes sitt felaktiga kommando kvällen innan.

Vid lunch ungefär frågar en av hans kollegor om han inte skall ta hand om sina tre kartonger som står vid postfacken. Vårn programmerare blir nu något konfunderad men uppsöker postfacken och finner där mycket riktigt tre kartonger fulla med hålkort. Varje kartong innehåller i storleksordningen 1500 hålkort. Programmeraren börjar nu ana vad som hade hänt

föregående kväll. Det kommando som han hade angivet var till för att kunna stansa en fil från QZ:s Tops-10-system på hålkort via Amdahl-systemet.

Nu backar vi i tid och rum och förflyttar oss till operatören för datasystemet och kvällen före. Operatören sitter och väntar på att klockan skall bli lagom mycket så att han får gå hem efter att ha kontrollerat att backup-rutinerna skall starta som planerat. Plötsligt börjar hålkortstansen låta i datahallen och operatören undrar då förvånat vad det är för någon tokf-n som stansar hålkort. Vi har här nu en ansvarskännande operatör så han kontrollerar vad som händer med stansen och fyller på hålkort vartefter dessa stansas.

Efter några minuter inser han att de tomma korten som han har uppe i datorhallen håller på att ta slut och nya behövs. Problemet med denna stans är att det inte går att hejda jobbet och skulle korten ta slut måste jobbet köras om från början. Mera hålkort finns i förrådet i källaren, c:a 150 meter längre bort. Våran kvälls-

arbetande operatör skynder sig därför ned i förrådet för att hämta mera hålkort och sedan tillbaka till stansen. Operatören hinner precis tillbaka till stansen och kan fylla på den innan korten tar slut. Operatören fortsätter att mata stansen med hålkort och plockar upp de stansade korten i kartonger. Stansningen fortgår och de ostan- sade kort är snart slut igen. Dags för en ny språngmarsch till förrådet nere i källaren. Operatören är nu inne på sin tredje kartong med hålkort men det räcker inte utan det var ju tre fulla kartonger som programmeraren hade fått så ytterligare en tur till källaren blev det. Kom nu ihåg att för varje kartong kort som var stansade så blev operatören mer och mer motiverad att hinna hämta en ny kartong annars skulle han vara tvungen att köra om allt från början.

Epilogen till denna historia är att våran programmerare iakttog mycket stor försiktighet när han närmade sig datorhallen de nästföljande månaderna, det var inte så populärt att motionera operatörerna på det sätt som han gjorde.

God of backup

by Greg Rose, published with due permission



HERE is a God whose prerogatives include backups. I don't know this God's name, but I know He (or She, or perhaps It, after all I wouldn't want to get sexist when talking about this God...) is definitely a vengeful God. Vengeful, angry, full of wrath, innovative in devising punishments and with a

warped sense of humor. Altogether, He (She, It) is not a fun Guy (Gal, Goo) to have around.

Take a case in point. This is a true story. I know the names of the individuals and companies involved, but I'm not going to tell you. Anyway, there is this computer vendor, let's call them Vendor

Inc., who sold a big, expensive computer to The Customer Corporation (say). This computer runs Customer Corp.'s absolutely vital, not to mention huge, corporate everything database.

This system and database had been installed for over ten years, and the Data Processing department of Customer Corp. had been doing its job quietly and happily for that time. Every six months, in addition to Customer Corp.'s regular backup schedule, they took a full-system image dump, ending up with more than 10 tapes. Then, being the professionals that they are, Customer Corp.'s Data Processing people sent the tapes to Vendor Inc. and paid a lot of money to have them loaded and verified. Vendor Inc. actually had to assemble a big enough machine and restore and run the system on it. Then the tapes were locked into a fireproof safe at Vendor Inc.'s headquarters in case they were needed.

Well, late in December, Customer Corp.'s mainframe lost a cabinet full of drives to a small fire. "No Worries," said the D.P. Manager, a Mr. Lamb. "We'll just get new drives under maintenance and reload from the backups!"

The replacement drives were installed within a day; after all, Customer Corp. was big and important, and they paid a lot for maintenance too. The tapes were retrieved from Vendor Inc.'s safe, and the reload commenced. As it happened, the major backup had been done in November, only a couple of weeks earlier.

Tape 7 was the problem. Tape 7 couldn't be read. To be slightly more precise, what was there could be read, but there wasn't anything there. Nothing at all.

It had been a brand new tape before the backup was done, of course, no expense was spared, and it had all the characteristics of a brand new tape now. It looked like some sort of procedural error had occurred, and one of the other tapes had been written twice, or some such irrelevant thing.

It was, of course, recrimination time. Poor Mr. Lamb was called up before the Board of Customer Corp., and asked to explain why the system had been down for two weeks. Fortunately, he was able to point the finger: Vendor Inc. was supposed to ensure that the backups were all there. The CEO of Customer Corp. called the CEO of Vendor Inc. and after the conversation had settled down a bit, asked why the backups had been useless after customer had spent the last 13 years paying (a lot) for them to be verified? "I'll get back to you," said Vendor Inc.'s CEO, with a sort of quaver in his voice.

Vendor Inc.'s CEO was noticeably more confident when he called back. "I have some bad news for you—I hope you're sitting down," he said to Customer Corp.'s Head Honcho. "When the tapes came last month there was a note attached. It said that you were fed up with paying so much to have the backups verified, after all they were never used, and you just wanted us to store them. It was signed by a Mr. B. Counter. Of course we did as requested. I'm sorry."

Mr. Lamb was fired for failing to verify that the verification had happened. Somewhere, I'm sure, there was the sound of Hysterical Laughter.

Many people think that "backup" is a noun, an object that you refer back

to when you need to. Others think that "backup" is a verb: the act of copying data. Any noun can be used as a verb. That's close, but here is the Truth.

A backup is a religious rite which propitiates the God of Backups.

It's obvious really. Everyone knows that if you take a good backup it will never be used. Even Murphy knew that. (Archives get used a lot. They are not backups. If you ever let your users know how easily you can restore files, they will delete important ones whenever they need temporary storage, then just ask you for them back. That's an archive, not a backup.)

The converse is also true. If you forget to take a backup, it isn't so bad, the God has other people he can Hassle, for a while anyway. But if you take a backup onto bad media, or have a power outage in the middle of one, or take a backup with a script that must be run in "/" when your current directory is "/tmp", or any of these sorts of things, well, you asked for it. Ker-Pow! The God of Backups wants his rituals done right, Or Else.

Just to establish my credentials (and show you how old I am), I'll give you another example. We'd been running Version 5 UNIX (no, that is not UNIX System V) for about six months when Version 6 came in. Doing a disk-to-disk copy of an RK05 using the block device took about 15 minutes. But V6 introduced raw devices!

We had three drives, not surprisingly numbered 0, 1, and 2. No tape, but removable packs. All backups were disk-to-disk. Well, we went to single user mode, put the system backup pack in drive 1, and started to dd from /dev/rk0 to /dev/rk1.

(I warned you about nouns and verbs, didn't I?) About halfway, our administrator stopped the copying, and said, "I'll just use the raw devices to make it faster." Of course, the moment the copying was interrupted, the God's beeper went off, and He began watching in fascination.

Our administrator quickly mknoded three times to create the raw devices. When the dd was restarted, this time on the raw devices, the lights on drives 0 and 2 (!) came on, and stayed on for a satisfying amount of time as entire tracks were copied. It took about thirty seconds before we noticed that the wrong drive was lighting up. The DecWriter console still showed the typo that had mixed up the drive minor numbers, but it was too late. The system disk backup contained the beginning of the current system disk, but the end of the old one. The end of the actual system disk was intact, but the beginning of it was clobbered by the front of the user disk. The situation might have been recovered at this point, since the whole system disk image existed, but in two pieces.

Of course, since the system was running off that system disk, it promptly crashed. Maniacal Laughter.

So when you are performing a backup, remember the proper sense of gravity in the situation. Your backup ceremony should be systematic. Label the tapes neatly, immediately after they come out of the drive. Store them properly. Don't interrupt a backup, and don't let anything stop you from finishing one. Establish a schedule of Worship, and stick to it.

And don't ever start giggling during a backup, or tell backup jokes...

