

Innehåller kallelse till SUGA:s föreningsstämma

AMIGA FORUM

www.suga.se
www.amigaforum.se

MARS 2016 #016

kurs
spel
mys
test

Grunderna i ARexx
Tanks Furry
5 klassiska demos
Extern GoteK

10 fiffiga
Amigatips

långtest:

Kan MiST ersätta en Amiga?

Om



Amiga Forum

det är något vi borde ha lärt oss om teknik så är det att det sällan är den bästa som vinner. Vi Amigaägare vet ju instinktivt att vår dator är/var bäst, att Intel sålde sjaskiga processorer och Microsofts diverse OS var både efterblivna, förvuxna och åt systemresurser som om det

inte fanns någon morgondag. Oj vad vi skratade åt detta.

Ändå så vann de. Överlägset. De knäckte inte bara Amiga utan även PPC, OS/2 Warp och BeOS trots att dessa var mycket bättre. Så här i efterhand så är anledningen solklar: man behöver inte vara bäst. Det räcker med att alla använder en. Där det finns mjukvara, dit går kunderna. Dit kunderna går, där skapas det mjukvara. På så sätt kunde de minst omtyckta teknologierna ta över världen. Det enda alternativet idag är Apples produkter. Apple lever på sin självbild om att vara för kreativt folk som

föredrar enkelhet. Men även de lutar mer och mer åt samma teknologiska lösningar som sina konkurrenter. Nyligen övergav de sitt eget Firewire till förmån för den sämre USB-standard. Det säger en del.

I år är det 30 år sedan den första Amigan kom till Sverige. År 2016 är datorerna och mjukvaran enormt mycket bättre än då, men samtidigt också väldigt mycket sämre. Som när saker och ting inte fungerar och programmen/hårdvaran vägrar berätta varför. Istället får man "Windows försöker hitta en lösning på ditt problem" tills det eller du ger upp. Det är då man innerligt längtar tillbaka till de utförliga manualerna, till en era när fel och felsökning var en accepterad del av datoranvändningen. Det var kanske inte kul, men man låtsades inte om att saker inte går sönder i alla fall.

Tänk på det nästa gång en PC laddar oändligt långsamt, någon fil blir korrupt eller ett nytt korts installation misslyckas. Detta är resultatet av att den sämsta tekniken vann.

Johannes Genberg

Innehåll

- | | |
|-----|----------------------------|
| 3. | Nyheter |
| 4. | SUGA-nytt |
| 6. | Kan MiST ersätta en Amiga? |
| 8. | Test: SCSI2SD |
| 10. | Gör om MOD till MP3 |
| 11. | Extern floppyemulator |
| 12. | Tanks Furry |
| 14. | Fem klassiska demos |
| 16. | Knep och knåp |
| 17. | Året var 1988 |
| 18. | Ännu fler Amigatips |
| 20. | AREXX-kurs |
| 24. | Boken om generation A |
| 26. | Efterlyses: en kul dator |



Omslagsbilden:
"Infestation" av Made.
(320x200, 16 färger)

Redaktör:	Johannes Genberg
Layout:	Iggly Drougge
Korrektur:	Joachim Melin

Medverkande i detta nummer:	Bent Floberg
	Ulf Urdén
	Jesper Odelius
	Jonas Hofmann

Amiga Forum ges ut av föreningen Swedish User Group of Amiga (SUGA). Tidningen utkommer 4 ggr/år. Artiklar och föreningsinformation skickas till redax@amigaforum.se. Deadline är den 15 februari, maj, augusti och november. Lösnummer säljs via suga.se/amigaforum/skaffa.html. Mejla redaktionen om din Amigaförening också vill prenumerera.

Nyheter från Amiga-världen

Mjukvara: WHDLoad släppt gratis

WHDLoad är ett programpaket som finns på snart sagt varje Amigaägares hårddisk idag då det låter en installera nästan alla gamla Amigaspel på hårddisk. Som en julkapp till Amigaägarna har huvudutvecklaren nu släppt programmet (men inte källkoden) gratis. Det betyder dock inte att utvecklingen lagts ned, utan nya versioner kommer även i fortsättningen.

www.whdload.de

Mjukvara: WHDLoad till CD32

CD32Load heter en klon av WHDLoad specialskriften för CD32. Det arbetar med samma .slave-filer som det riktiga WHDLoad, men laddar spelen från hembrända skivor istället. Naturligtvis begränsas antalet spel som går att köra av CD32:ans lilla minne och kompatibiliteten med "originalet" är inte heller helt fullständig.

eab.abime.net

Mjukvara: Google Drive till Amiga

En ny "handler" finns nu på Aminet för att komma åt Google Drive, både från AmigaOS 3.x och från MorphOS, OS4 och AROS. För att få skrivtöd krävs att man registrerar programmet för tio euro. Handlern är än så länge bara på betastadiet, men programmeraren arbetar vidare och har även utlovat stöd för Dropbox i en kommande version.

www.aminet.net

Mjukvara: Versionshantering med Git

Alla som programmerar i mer än AMOS känner nog till versionshanteringssystemet Git, som kan användas för att hålla koll på ändringar i källkod både på den egna datorn och i projekt med tusentals deltagare. Nu finns en (begränsad) version även för AmigaOS vid namn SGit (Simple Git).

aminet.net/package/dev/misc/sgit

Övrigt: Dave Needle har gått bort

Dave Needle var en av grundarna av Amiga och jämte Jay Miner huvudingenjör bakom Amigas specialkretsar. Efter att Amigan såldes till Commodore gick han vidare och var bland annat med om att ta fram Atari Lynx och 3DO ihop med RJ Mical. Needle avled den 20 februari.

Spel: Knightlore porterat till Amiga

En av de stora klassikerna från ZX Spectrum, "Knightlore" från Ultimate, har portats till Amigan. Portningen bygger på en översättning från Z80-maskinkod till C och har precis samma monokroma grafik som originalet – färggrafik kommer vad det lider.

retroports.blogspot.com

Hårdvara: Kickstart med flash till A500

Polackerna på Retro 7-bit ligger i. Nu har de byggt en ny kickstartswitch till Amiga 500 (och 2000) med flashminne så att man lätt kan byta kickstart utan att skruva upp Amigan. Man växlar kickstart genom att hålla inne ctrl-amiga-amiga i tre sekunder och nya kickstartar kan flashas in med ett litet program. Finns med flashminne på 512 kB och 1 MB.

retro.7-bit.pl

Hårdvara: Komponentvideomodulator

A520HD är namnet på en ny modul för att koppla Amigans RGB-utgång till komponentvideoringen på en modern tv (den med röda, gröna och blå RCA-kontakter). Den är kanske av störst intresse för användare i länder utan scart-ingång på tv:n, men en del hdtv-apparater har även dålig bild på scartkontakten. Hela konstruktionen är öppen "källkod" med både kopplingsschema och gerberfil (för etsning av kort) för fri nedladdning och kan byggas ihop av en entusiast med en enkel lödkolv och stadig hand.

ggglabs.us/node/983

Mjukvara: IFF-datatyp till MacOS

Om du kör Mac vet du säkert att man kan snabbtitta på bildfiler när man trycker på mellanslag i Finder – åtminstone om det är en JPEG, PNG eller TIFF. Dalton i Tulou har dock gjort ett instick till Quicktime så att man även kan titta på Amigans IFF ILBM-filer och även öppna dem i andra program.

github.com/dalton-tulou/ql-iff

Övrigt: Aminet letar gamla diskmagas

Sitter du på gamla diskettidningar, nyhetsbrev eller medlemsdisketter? Ladda upp dem till Aminet! Just nu försöker administratörerna bakom det största mjukvaruarkivet att få tag på så många publikationer som möjligt som annars riskerar försvinna.

www.aminet.net

Mjukvara: ProTracker utvecklas igen

Trackerprogrammets tracker, gamla hederliga ProTracker, har fått välbehövliga buggfixar och anpassningar för 2000-talets maskinvara. Eftersom tidigare vidareutvecklingar som version 3.15 och 3.61 aldrig blev särskilt populära är det den allra mest klassiska version 2.3 som nu vidareutvecklas.

www.16-bits.org/PT23E.LHA

Mjukvara: Demo på Gorky17 ute

En demoversion av actionspelet "Gorky17" till .OS4 finns nu för nedladdning.

www.hyperion-entertainment.biz

Rykten: Snart kommer nya IBrowse?

Alla som försökt använda IBrowse för att surfa på annat än Amiga-relaterade hemsidor de senaste åren har säkert känt av effekterna av att internet har vidareutvecklats sedan 2006 och att IBrowse inte har det. HTTPS har på senare tid blivit ett särskilt problem eftersom IBrowse inte fungerat med TLS 1.0 (ersätaren till SSL 3.0) och många sidor nu upphört med stödet för den äldre standarden. Om sidan även haft automatisk omdirigering från HTTP till HTTPS har den inte gått att ladda alls.

Nu har det dock visat sig att IBrowse inte nyttjat den fulla potentialen i OpenSSL 0.9.7g, som AmiSSL är baserat på. AmiSSL har nämligen TLS-stöd, det har bara alltid varit avstängt. Det går dock att slå på med följande shell-kommando:

```
setenv save AmiSSL/SSL_CLIENT_VERSION t1s1
```

I IBrowse inställningar behöver enligt uppgift SSLv2 och SSLv3 lämnas påslagna. Någon vidare hög säkerhetsnivå kan man inte vänta sig ändå, men kanske kan man lyckas besöka några fler sidor. Enligt nyheten på IBrowse hemsida kommer dock allt detta att lösas sig "i IBrowse 2.5 och en ny version av AmiSSL". Den hoppfulle kan därav sluta sig till att arbetet på IBrowse inte alls har avstannat, det fortskrider bara i samma takt som landhöjningen.

INFORMATION TILL MEDLEMMAR I SUGA

Kallelse till föreningsstämma i Swedish User Group of Amiga

Lördagen den 23 april klockan 17.00 är det åter dags för föreningsstämma i Swedish User Group of Amiga. Platsen är föreningens lokal på Folkungagatan 105.

Som föreningens högsta beslutande organ fastställer stämman budget och medlemsavgifter för kommande år samt väljer ny styrelse.

Om du vill kandidera till en post i styrelsen så skriv till Jonas Hofmann (jonas@suga.se) i valberedningen.

Dagordning:

- 1) Mötet förklaras öppnat.
- 2) Val av ordförande, sekreterare och två justeringsmän tillika rösträknare för föreningsstämman.
- 3) Fråga om kallelse till föreningsstämman skett på behörigt sätt.
- 4) Styrelsens verksamhetsberättelse.
- 5) Ekonomisk berättelse.
- 6) Revisorernas berättelse.
- 7) Beslut om ansvarsfrihet för styrelsen.
- 8) Beslut angående motioner som medlemmar stadgeenligt framfört till ordinarie föreningsstämma.
- 9) Beslut om medlemsavgift för inlett verksamhetsår.
- 10) Beslut om inventariemål.
- 11) Fastställande av verksamhetsplan.
- 12) Fastställande av budget.
- 13) Val av styrelse, revisor och valberedning.
- 14) Övriga frågor.
- 15) Mötet förklaras avslutat.

Mer om SCSI2SD

En uppföljning på testet i nummer 15

av Bent Floberg



min tidigare text om SCSI2SD hade jag vissa problem med läs- och skrivfel samt att Amigan hängde sig när den varit påslagen under en längre tid. Efter en del felökning så konstaterade jag att min Amiga 3000 hade en Western Digital scsi-krets version 3. I Amiga 3000 ska det vara minst ett version 4. Jag blev rejält snopen men hittade ett chips med version 4 hos Vesalia. Det bästa hade varit att få tag på ett version 8 men dessa är verkligen inte lätta att få tag på längre och då jag inte tänkter ha mer än en scsi-enhet så ska version 4 fungera bra. Efter byte av scsi-chips och ominstallation av SD-kortet

från scratch fungerade SCSI2SD helt strålande. Amigan fick jobba upp till 12 timmar utan ett enda fel eller hängning. Jag var så positiv att jag köpte nya nätverks- och usb-kort för nu var Amigan så stabil den kunde vara. Eller inte – läsfelen har kommit tillbaka men vi misstänker att det är PAL-kretsarna som blir för varma när de får jobba på gamla dar. Det verkar även vara något lurt med mitt dotterkort men problemet jag tidigare hade med SCSI2SD verkar i skrivande stund ha berott på att det satt ett buggigt scsi-chips i min kära Amiga 3000. 🌐

Rapport från SUGA

Det börjar bli ordning på torpet! En ny körning till brädgården ledde till en ny lagerhylla så att saker som tidigare förvarats under bord eller i travar på golvet undan för undan kunnat flyttas in i hyllor. Förvaringsproblemen har också bekämpats genom att föreningen införskaffat åtta nya possoboxar för disketter. Det sved i föreningskassan, men snyggt blir det i alla fall.

Efter att den nya hyllan byggts klart kunde vi också bygga en extra datorarbetsplats. Där står för närvarande en 16 MHz A3000 med högstrålande VGA-skärm. Det nya labbordet har försetts med en plattskärm med stöd för Amigans 15 KHz-lägen (en NEC 1970) på praktisk svängarm, men används mest till de pc-burkar som står där och agerar EPROM-brännare, logikanalysator och allmän testburk för hårdiskar och minnen. I bordets hyllor har programmeringsdokumentation flyttat in, medan det gamla labbordets hyllor vikts åt hårdvarudokumentation, komponenter och instrument. Där står även vår nya stereo, en hiskelig pjäs i silverplast från millennieskiftet som åtminstone inte tappar ljudkanaler som vår gamla 90-talsstereo gjorde.

Två gamla PowerMac G5 har fått skatta åt förgängelsen då lödningarna börjat släppa på grund av slarv på fabriken i kombination med hög värmeutveckling. Lyckligtvis har vi två sådana kvar för den som vill köra MorphOS. Tidningsarkivet har fyllts på lite till. Om du vill bli av med gamla eller nya Amigatidningar så hör av dig; vi har en stor hög med dubletter ifall du vill byta med oss också.

Du som får din tidning på posten har kanske också märkt att vi köpt nya färgband till skrivaren. Dessa bör räcka för många år framöver och även den mest skumögde brevbärare lär kunna läsa adressetiketten nu.

Styrelsen

Klon eller äkta vara?

Är en MiST en fungerande ersättare för en riktig Amiga?

av Johannes Genberg

NDet är bara att se sanningen i vitögat: dagens Amigaanvändare har en medelålder i övre 30-årsåldern eller äldre. För de flesta i den åldern kretsar livet kring jobb, familj och barn. Och då har man sällan tid eller råd med seriöst Amiganördande. Gammal hårdvara är inte helt lätt att hitta eller billigt att köpa. Det är betydligt lättare att expandera en Amiga så den fungerar med modern hårdvara såsom moderna plattskärmar eller CF-kort istället för pyttesmå IDE-hårddiskar, men inte heller det är billigt. Om du får onda ögat av maken eller hustrun varje gång du vill köpa en liten (men sjukt dyr) sak för era surt förvärvade pengar, vad ska du då göra för att behålla husfriden?

Då kanske MiST är ett alternativ för dig. Vi har redan beskrivit den i nummer 1 och nummer 14 av Amiga Forum vilket du kan läsa gratis som pdf (ladda ner på amigaforum.se) men de har varit mer tekniska genomgångar än recensioner av hur de faktiskt upplevs. Så jag kommer inte att ta upp mycket med det tekniska utan fokusera på om den duger som Amigaersättare.

Vad är en MiST? Det är en FPGA-baserad (Field Programmable Gate Array, eller programmerbara mikrochips) enkortsdator som var ursprungligen tänkt att vara en Atari ST-klon ("Mini ST"), men när det stod klart att Minimig-kärnan (en annan, äldre FPGA-Amigaklon vars kärna är öppen källkod) skulle fungera på MiST:en bytte namnet betydelse till A[MI]ga/Atari [ST]. Den kostar drygt 200 euro (220 med inbyggd MIDI). Utöver det behöver du en VGA-skärm, ett SD-kort, usb-tangentbord och -mus, vanliga datorhögtalare samt mikro-usb-laddare. Man kan också använda äkta Amigamöss samt Amigajoysticks/-pads. Även de flesta standard usb-joysticks/-pads fungerar också.

Vad motsvarar en MiST? Med den senaste betakärnan motsvarar den en smått uppdaterad Amiga 1200. Det vill säga 020, AGA samt upp till 24 MB Fast RAM. Och då är 30 % av FPGA:n fortfarande outnyttjad.

Att installera Amigakärnan är ganska enkelt men kräver en Kickstart, samt Workbenchdisketterna ifall man vill kunna använda hårddisk. Med den senaste versionen av kärnan kan man använda Amiga Forever-versionerna samt dess HDF-filer. Så saknar du det mesta blir MiST:en naturligtvis lite dyrare än 200 euro (plus frakt).

Ett till bekymmer är att MiST:en saknar sätt att installera filer utifrån (förutom ADF-filer). Usb-stickor fungerar inte (stöd kanske kommer senare) och något sätt att komma ut på internet finns inte heller. Av den anledningen kommer du att behöva använda dig av UAE för att den vägen överföra och installera filer på "hårddisk".

Jag använder mig av WinUAE. Med den kan jag skapa HDF-filer och installera program. Det gör man enkelt genom att stoppa in SD-kortet i en SD-kortläsare och låta UAE välja dess HDF-fil (om det finns någon. Annars får du skapa en och den måste heta `mist.hdf` för att automatiskt ladda i MiST:en). Via UAE kan man sen lägga till lokala lådor där saker du laddat ner från internet finns. Via dessa kan du sen installera i `mist.hdf`-filen. När du är klar behöver du bara stoppa in SD-kortet i MiST:en och starta. Fungerar det inte så klicka på F12 och ställ in så allt stämmer. Men om du inte bryr dig om att använda en hårddisk utan nöjer dig med ADF-filer behöver du inte använda UAE. Det räcker med att kopiera ADF:erna i valfri låda i SD-kortet och välja i MiST:en med F12. Du kan också som jag redan nämnt installera via ADF-filer, men filer på till exempel Aminet brukar komma i LHA-format och måste då läggas in i ADF-filer, vilket är bra mycket krångligare än att använda UAE.

Några program jag installerat är BetterWB, Directory Opus 4, DirWork 1.62, Hippoplayer, MagicWB samt WHDLoad (som nyligen blivit freeware). Plus en hel del spel och demos förstås.

Efter att ha testat i ett antal timmar så verkar de flesta Workbench-program fungera fint, även om det står klart att det finns problem med vissa. DirWork kraschar nästan hela tiden men Directory Opus liksom Hippoplayer fungerar fint. Jag lyssnade igenom hela min mod-samling utan problem. Att ändra inställningar i Prefs har också varit krashfritt. Så i det stora hela verkar det fungera som hos originalet. Men det är surt att min favoritfilhanterare Dirwork inte fungerar alls, vilket betyder att det borde finnas andra WB-program som inte heller fungerar. Dock kan vi anta att de är i minoritet.

Det viktigaste för en "casual" Amigaanvändare är förstås spel och demos. Och här stöter vi på fler problem.

Tack vare WHDLoad är det ganska enkelt att installera och köra spel från vilken Amiga som helst, även sådana som krånglar med vissa modeller. OCS/ECS-spel (A500, A600 etc) fungerar oftast vilket man kan tacka den ursprungliga Minimig-kärnan för. Några av mina favoritspel är till och med helt buggfria; som *Worms DC*, *Indy Heat*, *Benefactor* och *Ork Attack*. Särskilt den sista förvånade mig då det är skrivet i

AMOS och krånglar på de flesta emulatorer. Även de klassiska *Pinball Dreams* och *Fantasies* fungerar felfritt. *Damage* är en annan av mina favoriter och den fungerar med endast ett fåtal grafikbuggar närvarande.

AGA (A1200, A4000) är en annan femma. Här märks det att kärnan fortfarande är i beta-stadiet. Till exempel kraschar *Pinball Illusions* när man väljer bana, oavsett vilka inställningar man har ställt in. Däremot fungerar *Slam Tilt* helt felfritt, vilket kanske inte är så konstigt då man använt den som utgångspunkt när AGA-kärnan skrivits. PD-spelet *Roketz* lider av grafikbuggar men går att spela. *Super Stardust* fungerar men är så nedlusat av grafikbuggar att det inte är någon poäng att spela.

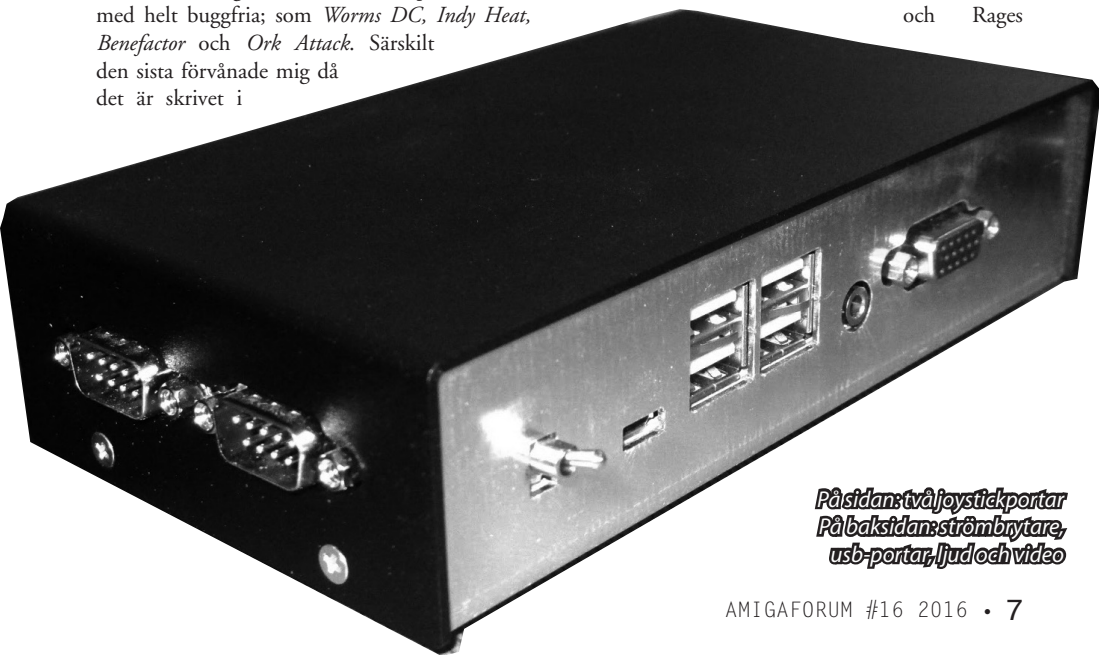
Även på demosidan är skillnaden mellan ocs/ecs och AGA tydlig. Det förstnämnda brukar

fungera bra även om grafikbuggar och timingproblem slinker in titt som tätt. Kända demos som Spaceballs

State of the Art och Mad Elks *Technological Death* fungerar helt okej även om den senare lider av just timingproblem. Ett flertal lite mindre kända demos jag testade fungerade också bra. På AGA-sidan är det värre.

Både Andromedas *Nexus 7* och Rages

Många A1200-spel lider fortfarande av grafikbuggar på gränsen till det ospelbara



**På sidan: två joystickportar,
på baksidan: strömbrytare,
usb-portar, ljud och video**

Maximum Overdrive 2 har så svåra timingproblem att man ger upp. En intressant observation är dock att det är oftare grafikbuggar på OCS/ECS-demonerna sällan i AGA-demonerna och att AGA-demonerna ofta lider av tajmingproblem medan OCS/ECS-demonerna sällan gör det. När det inte fungerar som det ska alltså.

Demos är dock notoriskt svåra att få att fungera, vilket är anledningen att de används som utgångspunkt för sådana här projekt. Demos är hack och utnyttjar inte bara hårdvarans styrka utan även dess buggar och fel. Dessa buggar och fel är sällan dokumenterade utan upptäcks av hackare och demoskapare när de experimenterar med hårdvaran. Så när MiST-kärnan (samt andra kärnor och emulatorer) förbättras så testar man det på demos för att ta reda på vad som inte fungerar och varför, för det är där alla problem märks som tydligast. Därför är det inte det minsta konstigt att Workbench fungerar riktigt bra, spel ganska bra och demos lite sisådär.

Vad är då domen? Som ni förstått så är MiST:en långt ifrån perfekt, men det är ingen som påstått att den är det heller. Den utvecklas fortfarande och kommer troligen bara bli bättre och bättre. Om du är någon som inte bryr sig det minsta om hur något fungerar så länge den bara gör det så är WinUAE fortfarande det överlägset bästa alternativet. Den ger inte bara en bättre upplevelse utan kan även emulera en mycket kraftigare maskin. Och det billigare dessutom. Till priset förstås att du måste starta (ärkefienden) Windows varje gång du vill leka med "Amigan". Är du Amigapurist och kräver endast äkta vara finns det inget att tillägga, men få får du nog leva med att det onda ögat förblir ständigt närvarande. Jag tänker inte hjälpa dig där.

Tycker du däremot att emulering är lite fusk och lite för långt ifrån en riktig Amiga så kan

en FPGA-lösning vara intressant. FPGA kallas för "hårdvaruemulering" för här översätts ingenting till främmande hårdvara i realtid. När FPGA:n väl är inställd att vara "Amiga" så är den enligt mjukvaran i allra högsta grad en Amiga. I teorin i alla fall, för man måste ju fortfarande skriva så hårdvaran fungerar som originalet vilket alla buggar och fel vittnar om är ett svårt uppdrag.

Nu kanske ni fått intrycket att MiST:en är en buggig historia. Men det är det inte. Visst finns det spel och program som inte fungerar helt eller delvis, men i det stora taget är jag riktigt nöjd med min MiST. Genom att undvika det som inte fungerar (och testa igen när en ny version av kärnan kommer) och hålla mig till det som fungerar bra är detta en mycket angenäm upplevelse. Vilket ju är vad man gör med en riktig Amiga också, för vi ska ju inte lura oss själva och tro att en sådan inte krashar. Ofta.

Jag har inte ägt en A1200 sedan slutet på 90-talet och den var lite kraftigare än vad en MiST klarar av, men efter att ha använt en i några veckor (de

MiST är långtifrån perfekt i nuläget, men blir bara bättre och bättre med tiden

stunder jag haft tid) så känns det hela nostalgiskt välbekant. Ifall du vill använda en FPGA för att rendera

3D i Imagine ska du köpa en FPGA Replay med det utannonserade o6o-kortet (och betala den betydande mellanskillnaden) men för nostalgi eller för att se hur mycket man kan pressa ut dess med dagens mått mått rejält blygsamma prestanda så är MiST:en helt klart högtintressant.

En till fördel är förstås att du även kan få en Atari ST, en Megadrive, en SNES och så vidare i samma maskin. Så är du en allätare har du plötsligt sparat en förmögenhet och en massa yta med detta köp. Det kan förstås aldrig ersätta den riktiga varan (vad den än må vara), men när barnen behöver nya skor kan en MiST vara det näst bästa, eller i alla fall kanske det enda alternativet till ingenting alls. 🌐



MiST

Läs mer: www.lotharek.pl

github.com/mist-devel

Pris: 200 € (Amigastore.eu)

Mod till MP3

Hur man konverterar moduler på bästa sätt

av Johannes Genberg

Har du en massa moduler du skulle vilja lyssna på i mp3-spelaren? Denna lilla guide ska berätta hur du konverterar dem till mp3-format på bästa sätt.

För detta behöver du tre program: UADE, SOX och LAME. Det första programmet är en Amiga-ljudchip emulator, det andra är en ljudfilskonverterare och den tredje är en MPEG-kodare.

Alla dessa tre är open source och finns till AmigaOS 4, MorphOS, AROS och Linux. Det bör även finnas för Windows och MacOS. De är lätta att hitta på de vanliga ställena eller via någon lämplig sökmotor. Däremot finns inte UADE för klassiska Amigor, men väl dess föregångare DeliTracker.

Först behöver vi en lämplig modul. Jag valde trackloading-låten *Syfilisflickan* (visst hade moduler fänigt fyndiga namn?) av Fajser för demot *Maximum Overdrive 2* av Rage.

Först ska vi konvertera modulen till RAW-format. Det gör vi genom att öppna ett shell, gå till UADE:s låda och skriva:

```
uade -P players/PTK-Prowiz -M  
mods:syfilisflickan.mod -outpipe 1  
>mods:syfilisflickan.raw
```

Jag tänker inte gå in på vad de olika kommandona gör – det är det man har dokumentationen till. Tänk bara på att denna inställning bara fungerar på Protracker-moduler. Om du vill konvertera ett annat format måste du ändra i "players". Mods: får du förstås ändra till den plats där modulen ligger.

Efter ett kort tag har vi en RAW-fil i samma låda. Denna måste konverteras till WAV. Det gör vi genom att närmast skriva i samma shell:

```
sox -u -b 16 -r 44100 -c 2 -s  
mods:syfilisflickan.raw mods:syfilisflickan.  
wav
```

Sox bör ligga i C: så du behöver inte byta låda. Du kan i teorin låta sox konvertera direkt till MP3 men resultatet blir enligt utsago inte helt bra, vilket är varför vi låter LAME ta hand om det sista steget. FLAC däremot tycker jag låter alldeles utmärkt så är det FLAC-filer du vill ha kan du skippa det sista steget.

Sista steget är att konvertera till en tillräckligt bra MP3:a, vilket LAME har en grundinställning för (precis som SOX borde det ligga i C: så du behöver inte byta låda):

```
lame -h mods:syfilisflickan.wav  
mods:syfilisflickan.mp3
```

Nu har vi en färdig MP3:a, redo att överföras till din MP3-spelare.

Men lyssnar vi på den färdiga låten märker vi att den både är väldigt kort och musiken spelas hårt i höger och vänster kanal. Detta är knappast optimalt för att lyssna i en mp3-spelare. Vi måste förlänga låten och mixa kanalerna.

Det kan man göra ganska lätt. Vi upprepar ovanstående steg, men skriver istället:

```
uade -P players/PTK-Prowiz -M -p 0.80  
-ne mods:syfilisflickan.mod -outpipe 1  
>mods:syfilisflickan.raw
```

Det "-p 0.80" gör är att mixa kanalerna mellan värdet 0.00 till 1.00. Lägst betyder att ingen mixning sker och högst betyder att allt ljud går jämnt ut i både höger och vänster kanal. Därför brukar ett värde mellan 0.50 eller 0.90 vara det bästa, men du måste själv experimentera med vilket värde som låter bäst för dig. Det brukar vara olika för olika moduler.

Det "-ne" gör är att upprepa låten i "evighet" men hos oss slutar den arbeta när den fått en 10 minuter lång loopad ljudfil. Det finns ingen inställning!
vänd!

ning för att säga åt UADE att bara loopa ett visst antal gånger. Du kan antingen låta låten bli 10 minuter lång, eller så kan du avbryta på känn med ctrl-c. Hur lång låten blir beror på hur snabb processor du har. På min Mac Mini på 1.5 GHz räknade jag långsamt till fem och fick då en drygt två minuter lång ljudfil. Även här får du testa dig fram.

En grej man kan störa sig på är att moduler ofta är gjorda för att loopas vilket gör att vår konverterade låt stannar tvärt. Det kan vi fixa med att tona ut ljudet i sox:

```
sox -u -b 16 -r 44100 -c 2 -s  
mods:syflisflickan.raw mods:syflisflickan.  
wav fade 0 0 7
```

Den första siffran i "fade" tonar in ljudet i sekunder och den sista tonar ut. Testa dig fram tills du får önskvärt resultat. Sen kör du LAME som ovan. Nu behöver du bara överföra filen till lämplig mp3-spelare. Glöm inte att ha på riktigt hög volym så att andra på bussen också får uppleva det härliga ljudet. 🌐

Extern Gotek

Kör floppyemulator som extern diskettstation

av Bent Floberg

Att ha en Gotek kopplad till sin Amiga istället för en vanlig diskdrive har många börjat uppskatta. Att bara lägga in adf-filer på ett usb-minne och slippa de allt vanligare läs- och skrivfelen som blir på gamla disketter med trötta diskettstationer. Det finns nya diskettstationer men det är svårare att hitta lämpliga nya disketter. Därmed ses Gotek som en skänk från ovan för retrofantaster.

Nu tillhör jag den underliga skaran som inte gillar att säga sönder sin Amiga utan föredrar att lägga Goteken ovanpå datorn. En variant vore att få tag

på en gammal extern diskettstation och stoppa i Goteken där, men det är ganska stor efterfrågan på externa diskettstationer. Nu finns dock möjligheten att köpa en enkel adapter med lämpliga kablar så Goteken kan läggas snyggt och tryggt ovanpå en A500. Goteken behöver inte ens byglas om till df1 utan den är bara att koppla in i adaptern.

Det hela fungerar ypperligt men det är några små saker som sänker betyget en aning. Ett är att om Goteken har en inbyggd högtalare som ska härma ljudet av en diskdrive så piper högtalaren även när den vanliga diskettstationen (df0) arbetar, typ diskdriveoljud i stereo. Det andra är att det inte finns någon kåpa som täcker själva adaptern.

Sammanfattningsvis menar jag att adaptern kan ses som en smidig lösning. Den gör det den ska och gör det till ett, i min uppfattning, rimligt pris. 🌐



En flatkabel och ett litet adapterkort är allt.

Extern floppyadapter M1

Amigastore.eu

Pris: 26,95 euro

WinUAE på Mac

Hjälp för dig som ledsnat på FS-UAE

av Ulf Urdén

De som kör Mac OS X som sitt huvudsakliga operativsystem och som ibland behöver emul(g)era en Amigamiljö har de senaste åren varit hänvisade till FS-UAE. Även om FS-UAE ger ett hyfsat Macanpassat gränssnitt så präglas den av en hel del märkliga designval. Bland dessa kan väl nämnas uppdelningen i två separata launcher/emuleringsfönster, att den som standard har utsträckt upplösning som gör bilden suddig, massvis med inställningar är gömda i konfigurationsfilen med mera.

Jag har alltid saknat WinUAE på Mac, och ibland därför kört det under en virtuell maskin. Lyckligtvis fick jag reda på att WinUAE faktiskt fungerar med hjälp av Wine. För den som inte vet vad Wine är så är det kortfattat ett projekt som emulerar Windows API, så att man kan köra Windows-mjukvara på andra plattformar.

Installationen är faktiskt ganska enkel. Först behöver man skaffa Homebrew, en pakethanterare för OS X. Gå till <http://brew.sh> och följ instruktionerna för att installera.

Sedan är det bara att öppna Terminal och skriva:
`$ brew install wine`

Wine är ett stort projekt med många beroenden. Det kan ta lång tid att kompilera beroende på vilken Mac man har. Sedan är det dags att ladda ner

WinUAE, gå till winuae.net, "Downloads" och välj "zip-archive".

Vi behöver packa upp WinUAE någonstans. Detta är bara ett förslag, det kan ligga var som helst:

```
$ unzip ~/Downloads/WinUAE3220.zip -d  
~/wine/drive_c/WinUAE
```

Nu kör vi igång:

```
$ wine ~/wine/drive_c/WinUAE/WinUAE.exe &  
C:\ motsvaras alltså av katalogen ~/wine/  
drive_c.
```

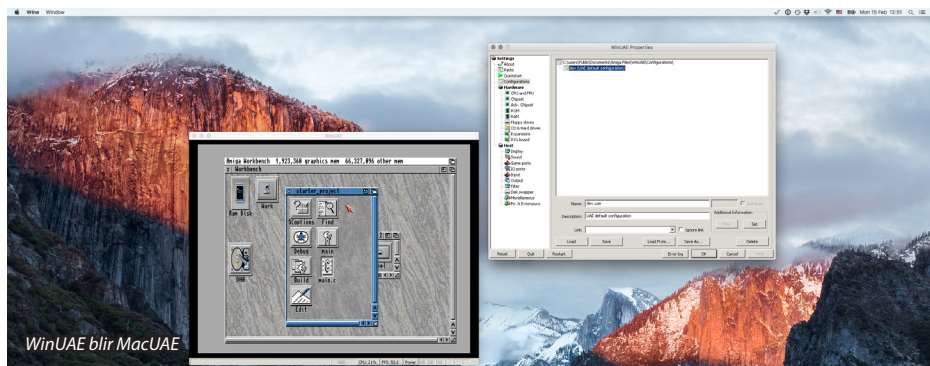
Sen är det bara att börja konfigurera!

Man kanske inte vill skriva ett långt kommando varje gång man skall starta WinUAE så jag har satt upp ett bash-alias som ligger i min `.bashrc` i stil med:

```
alias uae='wine ~/wine/drive_c/WinUAE/  
WinUAE.exe "C:\users\Public\Documents\  
Amiga Files\WinUAE\Configurations\dev.  
uae"'
```

Vill man piffa till det ytterligare går det säkert att skapa en ikon i Dock som startar igång WinUAE.

Visserligen har jag inte testat igenom alla funktioner, men jag har kört en hyfsat avancerad setup med grafikkort och hårddiskar på det här sättet utan några bekymmer alls. 🌐



TANKS FURRY



Söta djur och tunga stridsvagnar i nytt spel

av Iggy Drougge

Det är inte ofta det kommer nya Amigaspel nu för tiden, och desto mer sällan som de kommer på cd eller diskett med tryckt kartong och bruksanvisning. Därför är det extra spännande att testa Tanks furry, ett spel för en till fyra spelare där det gäller att spränga eller sprängas.

När jag fick förhandsexemplaret på mejlen och överfört det till Amigan klickade jag snabbt förbi introet, så exakt varför en råbarkad björn och krigsarrad varg (?) kör stridsvagn kan jag inte redogöra för. Det här är inte den typen av spel som behöver bakgrundshistoria, och skaparna i Project R3D ska få beröm för att de inte tvingar en att genomlida långgrandiga intron, liksom för att ha programmerat ett spel som funkar utan knussel både på en expanderad A4000 och en enkel A1200.

Tanks furry kan spelas även till fyra spelare med eller mot varandra, eller i ett kampanjläge för en till två samtidiga spelare. Jag har mest spelat i min ensamhet, men med en vapenbroder blir det dubbelt så roligt (åtminstone så länge både lyckas hålla sig lev).

Grafiken är konsolinspirerad, så till den grad att skaparna har beskyllts för att plagiera Nintendos *Advance wars*. Vi kan nöja oss med att konstatera att grafiken är inspirerad, men absolut inte kopierad, från Nintendo. Det är välanimerad och detaljerad pixelgrafik utan

att det blir plottrigt. Däremot är färgskalan mycket egen med dova färgtoner som om spelet färglagts i en batikverkstad i mitten av sjuttioalet. Ljudet kan mest beskrivas som funktionella puffar och smällar från granaterna.

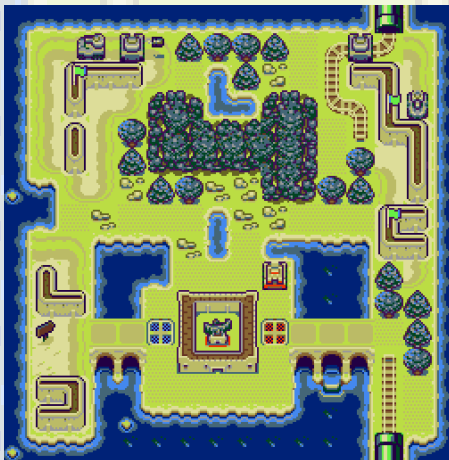
Det mesta på banan går att spränga med tillräckligt många skott, men så länge träden och pansarvärnen står kvar går de att använda som skydd, och det är detta som utgör mycket av spelmekaniken i *Tanks furry*. Den som spelat *Dynablast* eller *Bombberman* vet vad jag talar om, och det är möjligheten att leka

kurragömma med fienden eller lägga sig i bakhåll i skydd av en skogsdunge som gör det till ett lyckat flerspelarspel.

I en- eller två-spelarläge slås man istället av hur märkligt fiende-AI:n beter sig. Det är inte så att jag inte dör på var och varannan bana, men för det mesta känns det som om det mer var på grund av min egen klumpighet än genom fiendens list.

När jag försökte avsluta en spelsession genom att begå självmord var det nästan omöjligt att få fiendetanken att skjuta på mig trots att de hade mig mitt i siktet. Det är nog första gången jag stött på gentlemannamässiga fiender i ett spel, eller så är de bara väldigt fega.

Det finns flera olika fiendestridsvagnar med olika bra bepansring, och på senare banor även svävarstridsvagnar som kan teleportera. Inga av dem är dock särskilt smarta, så för det mesta är det den egna iveri som gör att man åker in



i ett fiendskott snarare än att fienderna själva lyckats göra någon smart manöver. Dock vet jag hur svårt det kan vara att göra bra AI som ska manövrera i trånga utrymmen, och gissningsvis är det de rutinerna som ligger bakom de ganska höga prestandakraven på vad som egentligen är ett rätt enkelt spel.

För att göra spelet mer omväxlande finns det diverse bonusar att plocka upp som snabbare motor och skott, specialare som fryser alla fiender, smarta bomber, teleportering och ett extraliv då och då. Dessutom dyker det då och då upp små gubbar som hoppar och vinkar för att bli räddade.

Ibland känns det som ett välpolerat åttabitarsspel med enkelt spelupplägg och dito grafik snarare än ett "äkt" Amigaspel, men det är inte bara en nackdel. Enkla spel är lätta att sätta sig in i och lätta att bara latta lite med, och då spelet har stöd för fyrspekaradapter kan *Tanks furry* få en given plats på datapartyn. Priset (gratis att ladda ner) är inte heller högre än att man kan förlåta några småskavanker. ☹



Tanks furry borde få en given plats på datapartyn

	Tanks Furry Plus Minus Systemkrav Pris Läs mer Enkelt och kul Segt ibland Höga systemkrav 1 MB chip, 68020 ca 70 kr http://amiga.net/pl
--	--

1 **BAMIGA SECTOR ONE & THE KENT TEAM:** *Utan namn* (1987)

Bamiga Sector One och The Kent Team i ett underbart cracktro. Det händer inte så värst mycket, men det är ett skönt tempo, vacker grafik, relativt mjuka animeringar och en underbar melodislinga. Jag har också alltid älskat "Amigabocken". Genomgående snygg typografi, vilket verkligen inte alltid var fallet med demoscenen. Bamiga Sector One var mina gudar, coolast och störst. De var från Belgien och samarbetade med flera olika grupper, de enda som jag brydde mig om var The Kent Team som verkade vara deras bästa "match" – när man såg något riktigt bra var det ofta BS1+TKT. TKT var från Kent i England, sas det i alla fall.



3 **UNIT A: INTERCEPTOR CRACKTRO** (1988)

Som BS1/TKT-demot ovan ett demo där det inte händer så mycket, men grafiken är sjukt snygg och jag gapade över det fullständigt oblyga att rakt av stjäla United Artists-logotypen. Skön stämning, snygga stjärnor, välavstämd musik och tre läckra mörkrets väktare tycker jag räcker för att utnämna det här till ett demo som står sig.



Fem klassiska

4 **CASCADE: CASCADE MEGADEMO 2** (1988)

Snygga, mjuka, studsande, krispiga scroller antyder riktigt bra hushållning med CPU = smart kod. Stundom grymt snygg typografi, allt handritat förstås. Några passager är direkt anskrämliga, men överlag läckert. Som fulsnyggast på ett coolt sätt vid 5'34 minuter och cirka en minut framåt – något slags nachochipsliknande scroll, och så bars. Jag älskar bars! Dessutom genomgående snygg musik från Paula.





2 SODAN & MAGICIAN 42: TECH TECH (1987)

Jag var så imponerad när jag såg det här demot för första gången att jag knappt kunde tala. Min Amigakumpan 5011 (Björn Olin) satt bredvid och sufflerade och berättade allt hopplöst jobbigt och svårt som Sodan & Magician 42 klarade av i demot, hur de maxade vad Amigan orkade och gjorde saker som ingen annan hade gjort förut. Vad visste jag? Assembler var fortfarande ett mysterium för mig även om jag gjorde tappra försök att begripa MOVE och slet med ett få en fungerande scroll. Allt i demot är fortfarande 2016 verkligen vidrigt snyggt. Samplingarna är fantastiska. Vektorgrafikdelen kan framstå som lite beige i dag, men var en av de passager som imponerade mest på mig då. Som kuriosa kan jag nämna att det 1996 i Stockholm startades en webbyrå som hette Tesch Tesch. På den här tiden hette det knappt webbyrå, snarare internetkonsult. Spray var the shit, men Tesch Tesch kom inte långt efter. Bröderna Måns och Johan Tesch låg bakom, så namnet har ju sin förklaring, men jag tänkte alltid att de måste ha inspirerats av Sodandemot och texten TECHTECH som ligger hela slutscenen igenom. Releasedatum 17 november 1987, snart trettio år sedan.

ska demos av Jesper Odelius



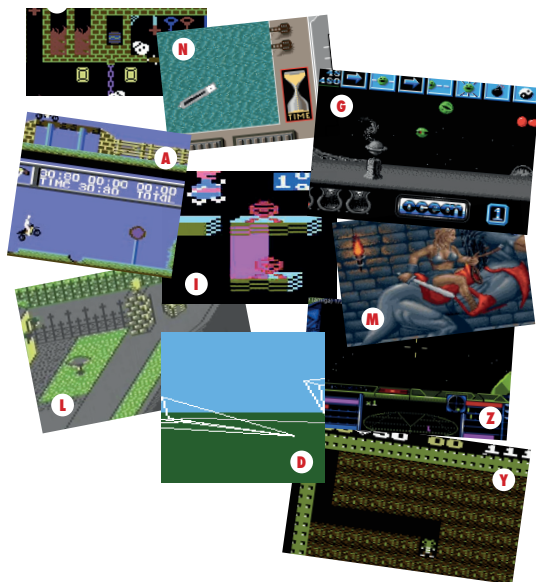
5 PHENOMENA: ENIGMA (1991)

Det här är ett klassiskt demo signerat svenska Phenomena, packat med knepiga saker. Stundom coolt och snyggt, stundom lite yxigare på den estetiska sidan (den svenska flaggan på flaggstång med guldknopp är avskyvärd), men lika fullt imponerande kod. "Olles boll" med 42 polygoner i fyra bitplan är kräkssnygg och Star Wars-scrollen är ett mästestycke. "Azatoth" bodde i Vällingby utanför Stockholm och jag i Bromma. Bara en sån sak: att ha Gud nästan runt hörnet.

KNEP OCH KNÅP

av Jesper Odelius

Para ihop rätt bokstav med rätt grå ruta. Du får ett ord om du har gjort rätt. För sen över bokstäverna 1,2,5,7, 3 i den ordningen i rutorna nedan till. Du ser om du gjort rätt. Obs att det finns en luring inbyggd...



- 1 ☐ Kikstart 2 (C64)
- 2 ☐ Sword of Sodan (Amiga)
- 3 ☐ Montezumas Revenge (C64)
- 4 ☐ Elite (Amiga)
- 5 ☐ Bounty Bob Strikes Back (C64)
- 6 ☐ Ports of Call (Amiga)
- 7 ☐ Wizball (Amiga)
- 8 ☐ Paperboy (C64)
- 9 ☐ Boulder Dash (Plus/4)

1	2	5	7	3
---	---	---	---	---

Före Amigan: Plus/4



Commodore Plus/4

Demo 1 (2011) <https://goo.gl/hWE5we>

Demo 2 (1993) <https://goo.gl/Kl57Od>



Commodore Plus/4 I Sverige såldes bara något hundratal exemplar av maskinen som marknadsfördes som en hemdator med inbyggd mjukvara för den som ville använda datorn till mer än spel. *Vem ville det?* Inbyggt som ROM fanns enkel ordbehandling, spreadsheet, ett grafitrarverktyg och en databasfunktion. Samtliga de små mjukvarorna dömdes ut av en enig datorpress och i USA kallades maskinen inte Plus 4 utan Minus 60, syftande på C64:an. Tillverkningen inleddes 1984 och lades ner 1986. Men maskinen lever fortfarande och det nyttillverkas såväl spel som stundom rätt coola demos.

ÅRET 1988 VAR...

Amigan är inte längre någon ny dator, utan talet går snarare om när nästa generation kommer, i fjärde delen i Bent Flobergs krönika om Amigans liv.

Nytt år, nya förhoppningar hos ägare av både Atari- och Amigadatorer. Commodores "krig" med Atari trappades upp men frågan var fortfarande vem som kom att vinna köparnas gunst. Försäljningen av Amigor ökade och enligt bland annat *Aboy Amiga* hade mer än 600 000 Amigor sålts till början av året och över 1 100 program hade publicerats. Även nya innovativa lösningar lanserades för att få en bättre ordning på skrivbordet (*Aboy #56*) även om de kanske inte tillverkades i några mängder (*se bild*).

I *Transactor #2* nämndes återigen ryktet om en ny Amiga, A3000, baserad på 68020 eller 68030. Valet skulle baseras på vad kostnaden för kretsarna var vid produktionsstart. Om valet blev 68020 skulle Amigan även få en 68851 MMU, i 68030 är MMU inbyggd. Oavsett ryktades det om att matteprocessorn skulle bli en 68882. Moderkortet skulle vara bestyckat med 2 eller möjligtvis 4 MB minne som standard. Dessutom skulle Amigan ha fler expansionsportar. Ryktet sade även att Amigan skulle klara 1024x920 utan interlace med en palett på 2 miljoner färger. En snabb uträkning visade att det skulle kräva enorma 1 megabyte per bitplan, men författaren räknade om till nästa nummer och kom fram till att 1 megabit per bitplan stämde bättre.

I *Info #18* ryktades om att ingenjörer hos Commodore hävdade att de snart är klara med en ny Denise till Amiga som kan visa 640x400 non-interlaced. Alla Amigor skulle kunna uppgraderas till den nya Denise när chipet blev klart.

Av någon anledning kändes det fortfarande viktigt att kunna köra C64- och C128-program på Amigan och i *Info #19* ryktades om att ett oberoende företag utvecklade ett Commodore 128-expansionskort. Kortet skulle ge ägare till Amiga 2000 (samt ägare till Amiga 500 och 1000 med lämplig expansionsadapter) att köra upp till 99,9 % av alla C64 och C128 program i ett amigafönster med full fart på emuleringen.

I *Info #21* nämndes att det höll på att bli ett



upplopp på ett uvecklarkonferens när deltagarna fick reda på att en viss Tim King skulle skriva ett nytt operativsystem kallat Helios för kort med transputerteknologi. Jag är inte insatt i vad Transputer egentligen innebar men det planerades även transputerkort som expansion till Amiga och saken var den att Tim tydligen programmerade den ursprungliga AmigaDOS. Även om det kanske inte lugnade den upprörda samlingen så menade de tyska ingenjörerna på mötet att "han har lärt sig mycket av sina misstag..."

Det ryktades om att en portabel Amiga var på gång. Det var Amigans systemmjukvaruguru Dale Luck som byggde en till sig själv. Enheten skulle bestå av ett A500-moderkort, 100 W Lambda-nät-del, 1 MB chipram, 2 MB fastram, nätverkskort, internt 2400-baudsmodem, 5 tums svartvit monitor, en 3,5-tumsdiskettstation och en 170 MB hårddisk. Hela härligheten skulle vara inbyggd i ett SX64-skål. Dale letade tydligen efter en 5-tumsmonitor i färg som skulle klara 80 kolumner, och uppskattat pris? Huj, cirka a 10 000 dollar. Vi ses 1989. 🌐

Tio Amigatips

Arbeta smartare med DOS och Workbench!

av Iggy "doktor DOS" Drougge

Avlusa startup-sekvensen

Om datorn hänger sig under uppstart kan det vara praktiskt att se vad som händer och var den hänger sig. Skriv **Set echo ON** i början av `S:Startup-Sequence` så skrivs alla rader som utförs ut på skärmen. Om du har OS 3.9 kan du istället skriva **Set interactive ON** så frågas du om varje rad ska utföras eller hoppas över.

Fokus på rätt fönster

När man växlar mellan skärmar får inte den nya skärmen fokus, utan man måste klicka i skärmen för att till exempel kunna skriva i det fönster man senast skrev i. Detta löser TWA (*Fish 781*) som återställer fokus till det senast använda fönstret vid skärmbbyte.

Bakgrundsbild i Workbench 2.0

Är du avundsjuk på 3.x-användarnas bakgrundsbilder i Workbench men vill/kan/vågar inte uppgradera till OS 3.1? Även OS 2.0-användare kan få bakgrundsbilder på sin Workbench tack vare `NickPrefs (util/wb/NickPrefs.lha)`.

Varför gurade datorn?

En gurumeditation är aldrig kul, men det känns kanske lite bättre om man åtminstone får veta vad de mystiska siffrorna betyder. Det talar `LastAlert (util/moni/DaLastAlert3.lha)` om. Lagg programmet i din `WBStartup` så får du en ruta med orsaken till senaste kraschen när datorn startat om.

Hårdvaran styr skriptet

Med `ShowBoards (util/libs/BoardsLib.lha)` kan du inte bara visa vilka kort och vilka chips som finns i din dator, utan även utnyttja det för att styra dina skript. Om du anger ett autoconfig-ID som argument till `ShowBoards` sätts nämligen returkoden i Shell beroende på om kortet finns i systemet eller inte. Kör kommandot en gång för att se vilket ID dina kort har – ett tillverkar-ID och ett kort-ID. Med `ShowBoards ALLKNOWN` visas alla kort som `boards.library` över huvud taget känner till. Till exempel har ett Hydra-ethernetkort tillverkar-ID 2121 och kort-ID 1. För att se om kortet finns i datorn skriver du då `ShowBoards MID=2121 PID=1`. Om kortet hittades blir returkoden 0 (kommandot lyckades), annars 5 (varning). På så vis kan du ange om din TCP/IP-stack ska startas (eller med vilket kort den ska startas) genom dessa rader:

```
ShowBoards MID=2121 PID=1
If WARN
```

```
Echo "Nätverkskortet hittades ej!"
```

```
Else
```

```
Echo "Hittade Hydrakortet. Startar AmiTCP."
```

```
Startnet
```

```
Endif
```

Detta banala exempel visar inte hur användbart detta tricks kan göra. Med en finslipad startupsekvens kan samma installation användas på flera datorer, eller samma startupsekvens användas på en dator med acceleratorm både på- och avslagen genom att kopiera in bibliotek och ställa assigns för rätt cpu-version eller hoppa över stora och tunga program när acceleratorm är deaktiverad. På samma vis kan man utan vidare flytta över sitt system till UAE utan att drivrutiner börjar klagas.

Hämta aktuellt processnummer

Om man skriver ett shellskript som skapar temporära filer kan det vara nödvändigt att ge filerna ett unikt namn så att det inte uppstår krockar om man råkar köra samma skript i flera shellfönster samtidigt. Det aktuella processnumret kan användas för att skapa det filnamnet, och kan hämtas antingen ur den lokala variabeln `$Process` eller ur specialvariabeln `<$$>`. Skript som använder `<$$>` måste börja med en `.KEY`-rad. Klammrarna kan bytas ut mot andra tecken med `.BRA` och `.KET` och dollartecknet med `.DOL` eller `.DOLLAR`.

Flytta fönster snabbt i OS4

Håll nere `ctrl-alt` så kan du flytta ett fönster genom att klicka och dra med musen var som helst i fönstret.

Publika skärmar

AmigaOS arbetar som bekant med virtuella skärmar, och sedan version 2.0 finns dessa av två slag: privata och publika. Publika skärmar är tillgängliga för alla program – paradexemplet är Workbenchskärmen som alla program som inte öppnar sin egen skärm lägger sina fönster på. Många program kan dock sägas åt att öppnas på vilken publik skärm som helst, medan andra öppnas på standardskärmen, som vanligtvis är Workbench. Med `PubChange` (*Fish 804*) kan du ställa in den främsta skärmen som standardskärm, så att t ex commodities öppnas där när du trycker deras snabbknapp.

Ändra standardverktyget

När man installerar ett program eller öppnar andras filer har de ofta fel program (eller fel sökväg) inställt som standardverktyg. Med `Deft` (*Fish 739*) kan man söka igenom en hel låda eller hel partition efter dokument som vill öppnas med t ex `AmigaGuide` och ändra dem till att öppnas med `MultiView`.

Systemvariabler

AmigaOS och dess program arbetar mycket med systemvariabler, till exempel lagras systemversionen i de globala variablerna `$Kickstart` och `$Workbench`. För att visa dem kan du skriva `Echo $Workbench`, där dollartecknet anger att det är en variabel och inte bara ett ord vilket som helst. Stora eller små bokstäver spelar ingen roll i variabelnamn. Ett alternativ till `Echo` är `GetEnv`, men då utelämnas dollartecknet eftersom det kommandot endast arbetar med variabelnamn och inget annat.

Det finns både globala och lokala variabler. Lokala variabler gäller bara i det aktuella shellfönstret medan de globala kan kommas åt från hela systemet. Lokala variabler sätts med `Set`, hämtas med `Get` och rensas med `Unset`. Om du inte anger något argument till `Set` listas alla lokala variabler. Här kan vi se att varje shell har tre lokala variabler. `$RC` anger svarskoden (felkoden) för det senaste kommandot, `$Result2` innehåller den sekundära felkoden som vissa program lämnar ifrån sig och `$Process` innehåller numret på den aktuella shellprocessen. Den kan vara användbar i shellskript om man behöver skapa temporära filer med ett unikt identifikationsnummer.

Globala variabler sätts med `SetEnv` och hämtas med `GetEnv`. Precis som `Set` kan `SetEnv` lista alla globala variabler. De globala variablerna kan också kommas åt som filer i `ENV:`. Listar du innehållet i `ENV:` ser du att det innehåller samma rader som `SetEnv`. `ENV:` kan också innehålla underkataloger, så `GetEnv Wordworth7/WwFonts` hämtar innehållet ur filen `ENV:Wordworth7/WwFonts`. Mycket av innehållet i `ENV:` är dock inte rena textfiler utan IFF-filer och annat, så ofta får man inte ut så mycket vettigt ur programspecifika variabler.

I de fall en variabel innehåller ett filnamn eller en sträng med mellanslag är det en bra idé att innesluta variabelnamnet i mäsvingar, så att t ex `Delete {sminfil}` i ett skript fungerar som det ska och inte stannar för att det försöker komma åt fel filer.



ARexx för nybörjare

En introduktion till ett enkelt men kraftfullt språk

av Iggy Drougge

Om du följt med i AmigaDOS-sidorna i de senaste numren har du kanske lagt märke till att AmigaOS har rätt många vårtor, inte minst i Shell. Förklaringarna är flera. Dels kom AmigaDOS som en främmande fågel från en utomstående utvecklare och gjordes på rekordtid, dels förstod sig nästan ingen på Commodore på hur källkoden riktigt hängde ihop. Men en annan förklaring är att AmigaOS fick ARexx i och med version 2.0. ARexx ersätter inte AmigaDOS på något sätt, men kan sköta mycket av det som DOS är dåligt på, och mer därtill. Det är ett programspråk så enkelt att till och med din mamma kan lära sig det, och samtidigt så kraftfullt att det kan göra mer än något shellskript kan. Därför är det på tiden att du lär dig ARexx så att du verkligen kan ta kontroll över din Amiga.

```
/* Mitt första program */  
SAY 'Vad heter du?'  
PULL namn  
SAY 'Välkommen till ARexx,' namn'!
```

Spara texten någonstans och kör den med "RX filnamn". Av hävd avslutas ARexx-filnamn med ".rexx", men det är inte tvingande.

Programmet består av fyra rader. Den första är en kommentar och utförs inte, men varje program *måste* börja med en kommentar för att visa att det är ett ARexxprogram och inte ett shellskript eller något annat. Kommentarer börjar med /* och slutar med */ precis som i C.

Nästa rad innehåller det första kommandot. SAY skriver bara ut en sträng på skärmen. Strängar kan inneslutas med enkla eller dubbla citattecken. Vill du skriva ut ett dubbelt citattecken kan du innesluta det i enkla (""") och vice versa.

PULL läser in en rad från användaren och lägger i variabeln namn. Vad variabler är förutsätter jag att du vet, och i denna kurs går vi bara in på dem

i den mån de skiljer sig från andra programspråk. Vi kan nöja oss med att konstatera att variabler i ARexx är typlösa och lika gärna kan innehålla tal som strängar.

Den sista raden visar en av egenheterna i ARexx. I andra språk behöver man pussla ihop strängar och variabler med specialtecken och funktioner som gör koden både längre och mer svårläst. I ARexx sätts de istället ihop genom att... sätta ihop dem. Om man svarar "Iggy" på frågan får man alltså svaret "Välkommen till ARexx, Iggy!". Med ett mellanslag mellan komponenterna sätts ett mellanslag in, och utan det sätts de ihop utan mellanslag.

Här skriver jag alla kommandon med versaler, men ARexx bryr sig inte om sånt. Jag gör det bara för att programmet ska vara tydligare och för att min texteditor (TurboText) gör det automatiskt.

Loopar och block

Alla loopar i ARexx börjar med DO och slutar med END. DO inleder samtidigt ett block, motsvarande måsvingarna i C och Java eller diverse konstruktioner som If/Endif och For/Next i Basic. Genom att ange argument till DO ändras beteendet på loopen. Till exempel kan vi skriva:

```
DO FOREVER  
  SAY 'Stoppa mig om du kan!'  
END
```

Och få ett program som aldrig stannar. Det låter farligt, men är ofta en bra huvudloop i ett program. Vi kan också få loopen att bara köra ett antal gånger:

```
DO 10  
  SAY 'Samma text tio gånger i rad'  
END
```

Likasa kan vi få for-loopar med räknare på följande vis:

```
DO ggr=1 TO 10  
  SAY 'Räknar till tio...' ggr  
END
```

```
DO ggr=2 TO 10 BY 2
  SAY 'Jämna tal mellan 1 & 10:' ggr
END
DO ggr=5 to 1 BY -1
  SAY ggr 'kvar...'
END
```

Sedan har vi DO UNTIL och DO WHILE som upprepar sig så länge ett villkor är uppfyllt:

```
ggr=0
DO UNTIL ggr=10
  ggr=ggr+1
  SAY ggr 'gångar...'
END
ggr=0
DO WHILE ggr<10
  SAY ggr 'gångar...'
END
```

Man kan också kombinera dessa element:

```
DO ggr=1 TO 10 UNTIL klar=ja
  SAY ggr 'gångar...'
  IF ggr=5 THEN klar=ja
END
```

Denna sista loop körs bara fem gånger, eftersom variabeln klar sätts till "JA" när ggr blivit 5. Här ser vi dels hur man testar villkor med IF...THEN, dels en egenhet i hur AREXX behandlar variabler. Om en variabel aldrig tilldelats ett värde innehåller den sitt eget namn med versaler. I början av programmet innehåller alltså variabeln klar ordet "KLAR" och variabeln ja ordet "JA". Detta är rätt unikt för AREXX och kan både hjälpa och stjälpa. I detta fall tar vi hjälp av det för att hålla programmet så kort som möjligt, men det blir lätt fel om man glömt att till exempel sätta en räknarvariabel innan man börjar räkna på den eftersom ggr+1 blir väldigt fel om man försöker addera talet 1 till strängen GGR. Nu kan vi titta på fler sätt att styra programflödet med IF.

```
DO UNTIL dator=amiga
  SAY 'Vilken är den bästa datorn?'
  PULL dator
  IF dator=amiga
  THEN SAY 'Helt rätt!'
  ELSE SAY 'Fel. Försök igen.'
END
```

I början av programmet innehåller variabeln dator ordet "DATOR", men fylls sedan med det

ord användaren matar in. Värdet jämförs med variabeln amiga, som tills vidare innehåller ordet "AMIGA". Här spelar det ingen roll om användare skriver "Amiga", "amiga" eller "amiGA", för PULL omvandlar det inmatade ordet till versaler. I själva verket är PULL en kortform för PARSE UPPER PULL, och vill vi behålla små och stora bokstäver byter vi ut raden mot PARSE PULL (utan UPPER, d v s versaler). I detta fall vore det bara onödigt eftersom vi då skulle kräva att användaren stavade "Amiga" på helt rätt sätt. Vi hade också kunnat skriva IF dator='AMIGA', men eftersom variabeln amiga redan innehåller just strängen "AMIGA" är det onödigt. Däremot kan det hända att någon lömsk person smyger in en rad i början av programmet: amiga='ATARI'. Då hade utfallet blivit helt anorlunda...

Det finns två kommandon som direkt styr flödet i en loop: ITERATE och LEAVE.

```
DO FOREVER
  SAY 'Vilken är den bästa datorn?'
  PULL dator
  IF dator=amiga THEN DO
    SAY 'Helt rätt!'
  LEAVE
  END
  ELSE DO
    SAY 'Du gissade fel. Försök igen.'
  ITERATE
  END
  SAY 'Den här raden skrivs aldrig ut.'


```
END
SAY 'Du kom ut ur den oändliga loopen.'
```


```

ITERATE hoppar upp till början av loopen, medan LEAVE hoppar ut ur den. Vi ser här också hur DO/END anger programblock och hur en DO kan ligga inuti en annan DO.

Om man vill utvärdera flera olika svarsalternativ kan man visserligen lägga en ny IF i sin IF-sats så att man får något som ser ut som IF...THEN...ELSE IF...THEN...ELSE IF...THEN...ELSE IF...THEN...ELSE men det blir onödigt svårsläst, så då finns det en bättre konstruktion i form av SELECT. Det påminner om switch/case i C, men är kraftfullare eftersom varje WHEN-sats kan genomföra sin egen kontroll.


```

SELECT
  WHEN dator=atari
    THEN SAY 'Knappast!'
  WHEN dator=amiga THEN DO
    SAY 'Helt rätt!'
  LEAVE
  END
  WHEN dator='PC'
    THEN SAY 'Glöm det!'
  OTHERWISE SAY 'Den har jag aldrig
hört talas om!'
  END

```

Parsning

Nu när vi loopat på en rad sätt är det dags att ta itu med ARExx allra kraftfullaste kommando: PARSE. Vi stötte redan på det i form av PULL, som är en kortform av PARSE UPPER PULL. PARSE PULL läser en rad från användaren (eller närmare bestämt från stdin) och lägger i en variabel, men det PARSE egentligen gör är att behandla indata. PARSE VAR läser in en variabel istället, och PARSE ARG arbetar med argumenten till ett program eller en procedur. Så om PARSE PULL bla läser in användarens inmatning till variabeln bla läser PARSE VAR tjoff bla in innehållet i variabeln tjoff och lägger det i variabeln bla. Det är förstås helt meningslöst eftersom man lika gärna kunde skriva bla=tjoff, men nu ska vi se vad mer man kan göra:

```

SAY 'Ange ditt fullständiga namn:'
PARSE PULL fnamn enamn
SAY 'Ditt förnamn är alltså' fnamn
'och du heter' enamn 'i efternamn.'

```

Detta enkla exempel läser in en sträng från användaren och delar sedan upp den (vid första mellanslaget) på två variabler fnamn och enamn. Om man svarar "Kalle Anka" blir fnamn="Kalle" och enamn="Anka".

Att göra något liknande är betydligt längre och krångligare i andra programspråk, utom de som kan hantera regexpar, i vilket fall det blir kortare men fortfarande krångligare. Säg att man har ett program som läser rader i en en medlemsdatabas som ser ut så här: medlemsnr förnamn efternamn telefonnummer adress

Om databasen har fast längd på fälten – till exempel 5 tecken för medlemsnummer, 40 tecken för adressen och 20 tecken för övriga fält – låter sig en rad i listan läsas in så här:

```

rad=ReadLn(databas)
PARSE VAR rad medlemsnr +5 fnamn +20
  enamn +20 telnr +20 adress +40 .

```

Då läser ARExx in de första fem tecknen i variabeln rad till variabeln medlemsnr, sedan följande tjugo tecken i variabeln fnamn och så vidare. Det sista +40 . skulle man kunna hoppa över eftersom PARSE fortsätter att läsa till slutet på källvariabeln och lägger allt som återstår i den sista målvariabeln, men om raden innehåller andra onödiga fält efter adressfältet ska de inte läsas in i adressvariabeln, och därför säger vi åt PARSE att bara läsa in 40 tecken där och kasta bort resten, vilket anges med en punkt. Detta är en mer avancerad mall eller "template" till PARSE-kommandot.

Låt oss säga att databasraderna istället följde det vanliga CSV-formatet (kommaseparerad lista) som man till exempel kan exportera från Excel. Då ser mallen ut så här:

```

PARSE VAR rad ',' medlemsnr ',' fnamn
',' enamn ',' telnr ',' adress ',' .

```

Man kan även ange källvariabeln som målvariabel i mallen. På så vis kan man göra iterativa eller rekursiva funktioner där ARExx läser en källvariabel bit för bit genom att stycka av bitar av källvariabeln och lägga tillbaka återstoden. I annat fall förblir källvariabeln oförändrad av att "parsas".

```

massaord='Här är en mening med en
massa ord.'
DO WHILE invariabel ~= ''
  PARSE VAR massaord ord1 massaord
  SAY 'Hittade ett ord:' ord1
  SAY 'Återstående ord:' massaord
END

```

Observera att ~= betyder "ej lika med", så att loopen fortsätter så länge invariabeln inte är tom. Man hade förstås också kunnat skriva DO UNTIL invariabel= ''.

Förutom att ange hopp framåt i mallen med + kan man ange fasta positioner att läsa från. I exemplet med databasen med fasta fält hade man istället kunnat använda sig av mallen PARSE VAR rad 1 medlemsnr 5 fnamn 25 enamn 45 telnr 65 adress 105 .

Man kan även ersätta fasta tal med variabler. De skrivs då in i mallen med likamedtecken och parenteser. Så här skrev jag i ett program som skulle läsa in en textfil och ersätta alla &-tecken med XML-motsvarigheten & för att filen skulle godtas av Adobes stränga XML-läsare:

```

do while ~eof(infil)
  inrad=readln(infil)
  och=1
  do while index(inrad,'&',och)>0
    och=index(inrad,'&',och)
    parse value inrad with prefix
  =(och) +1 suffix
    inrad=prefix'&':suffix
    och=index(inrad,':',och)
  end
end
end

```

Den yttre loopen läser in rader från infilen tills den når *end-of-file*. För varje rad sätts först räknaren "och" (som räknar var första &-tecknet finns) till 1 (början av raden) och kollar sedan med funktionen `index(sträng,sökord,startposition)` om det finns några &-tecken i inraden. Så länge det finns pågår den inre loopen som först lägger positionen för det första &-tecknet i variabeln `och`. Den variabeln används sedan med `PARSE` som positionsmarkör följd av ett hopp en bokstav framåt så att all text fram till `och`-positionen läggs i variabeln `prefix`, &-tecknet kastas bort och all text efter det läggs i variabeln `suffix`. Sedan läggs variablerna `prefix` och `suffix` ihop med strängen `&`; emellan. Efter det hoppar och fram till positionen för det avslutande semikolonet och loopen börjar om så att den letar efter nästa &-tecken från den position och står i, alltså efter semikolonet.

Observera att vi här skriver `PARSE VALUE WITH` istället för `PARSE VAR`. Skillnaden är oväsentlig här, men vi hade kunnat skriva `PARSE VALUE readln(infil)` i ett annat exempel. `PARSE VAR` arbetar alltså bara med variabler, medan `PARSE VALUE` kan arbeta med alla uttryck, inklusive funktioner. Ett ganska roligt tricks är att byta innehållet på två variabler med `PARSE a b WITH b a` för att byta innehåll mellan två variabler utan att använda någon temporär variabel som mellanlagring.

Kommunikation och portar

ARExx är nog mest känt för som kommunikationspråk mellan olika program och internt skriptspråk i programmen. Alla ordentliga Amigaprogram har en ARExxport som kan användas för att automatisera dem eller utbyta information med andra program. För eget bruk har jag till exempel gjort ett ARExx-makro till Turbotext för att skicka in ett Blitz Basic-program till Blitz egen editor TED och kompilera det automatiskt och ett annat för att

mappa om färger på spritar i Personal Paint och göra översiktbilder med alla spritebilder. Nyckeln är att peka om ARExx så att det skickar kommandon inte till ARExx själv, utan till ett programs ARExxport. Vi kan först se om det finns några portar öppna i din dator med denna rad i ett shellfönster:

```
RX "SAY show(ports)"
```

Alla Amigor har åtminstone några portar öppna så länge REXxMast är igång, allra minst ARExx och COMMAND. Den förra är ARExx egen port, och den andra är AmigaDOS port. Porten COMMAND gör inte så mycket mer än att utföra de kommandon man skickar till den, men det duger för ett exempel:

```
SAY 'Vilken fil ska jag radera?'
```

```
PARSE PULL filnamn
```

```
ADDRESS COMMAND 'Delete' filnamn
```

Ett opraktiskt exempel, men det innebär att mer komplexa ARExxprogram kan ta hjälp av alla kommandon i C: för att hantera filer och så vidare.

ARExxportar är rätt enkla saker. De tar emot kommandon och reagerar på dem. Varje program har sina egna kommandon som brukar beskrivas i programmets dokumentation. Vill man veta vad kommandot fick för resultat tittar man i de två variablerna `RC` (resultatkod eller felkod) och `RESULT` (returvärde). Du kan behöva aktivera dessa med kommandot `OPTIONS RESULTS`. Säg att du anger en fil som inte finns i det ovanstående programmet. `RC` kommer då att vara 5. En nolla betyder att inga fel uppstod. `RC`-koderna är de samma som i AmigaDOS, d v s att en femma är varning, en tia ett fel och tjugo totaltnas. Vissa program definierar sina egna `RC`-koder. Koden använder du i dina skript för att styra programförloppet, medan `RESULT` innehåller portens egna svar, till exempel ett filnamn från `DirOpus` eller en textrad från `CygnusEd`.

Med `ADDRESS` anger du vilken port du ska prata med, och portarnas namn skiljer (till skillnad från allt inuti själva ARExx) mellan stora och små bokstäver så att stavningen måste stämma helt. Anger du `ADDRESS 'port'` på en egen rad pekas ARExx om till den porten så att alla kommandon som ARExx inte själv känner igen skickas dit. Anger du det istället i början på en rad följd av ett kommando gäller den angivna porten bara för den raden. Detta är användbart då man ska knyta ihop två olika programs portar. Men mer om det i nästa nummer. Om du kör fast eller har några önskningsar inför nästa del är det bara att slänga iväg ett mejl till iggy@suga.se! 🌐

Generation Amiga

Nu blir den av – boken om hur Amigan formade en generation

av Johannes Genberg

För två år sedan skrev Jimmy Wilhelmsson boken *Generation 64*, om unga svenskar som växte upp med hemdatorn Commodore 64, en soffbordsbok formgiven av Kenneth Grönwall. Nu är de båda tillbaka och arbetar tillsammans igen på en bok om Amiga-generationen – och svarar på lite av våra frågor om sitt bokprojekt.

Hej Jimmy, det blir alltså en Amiga-bok – vad fick er att bestämma er för att skriva den?

– Ska jag vara ärlig så togs inte detta beslut förrän en bit in på 2016. Jag skrev under större delen av förra året en bok om rollspelsförlaget Äventyrs-spel, som bland annat gav ut Drakar och Demoner, och hade även ett finger med i översättningen av *Generation 64* som tog vid i höst. Så mina tankar låg inte direkt på en Amiga-bok, men var jag än begett mig så har folk kommit fram till mig och undrat om jag tänker ta mig an Amiga-generationen, så man kan väl säga att deras tjat lönat sig (skrattar).

Nej, men skämt åsido – jag hade några andra bokprojekt på gång, men fick under februari fler och fler idéer kring just en Amiga-bok. Nu tänker jag att jag måste avsluta det jag påbörjat, eftersom Amiga 500 varit en så stor del av min egen uppväxt känner jag att jag måste få det gjort. Och 2017 känns som en bra tidpunkt, 30 år efter släppet av Amiga 500.

Kommer det att bli en bok liknande *Generation 64* eller något helt annat?

– Svårt att säga, vi befinner oss just nu i planeringsstadiet – det roligaste som finns, ett stadium där allt fortfarande kan hända och inga ramar eller begränsningar har ännu dykt upp. Men arbetsnam-

net på boken är *Generation 500*, det kan vi ju säga – och tanken är nog att ta vid där *Generation 64* slutade, det finns en naturlig transition mellan de båda datorerna i mitt liv, och säkert också i andras.

Däremot är boken om Amiga-generationen en helt annan vad gäller utgångspunkt och vinkel. Om Commodore 64 var en mysig, kantig och inramad burk som folk samlades kring i stora spel- och democommunities, så blev Amiga 500 snarare den ultimata framtidsdatorn som visade vartåt utvecklingen skulle gå. Vuxna harvade på med sina IBM och Apple-datorer, men vi kids satt på våra pojk- och flickrum och byggde den digitala revolutionen på riktigt.

Amiga 500 gav många av oss en första inblick i operativsystem, mus, mängder med samtida färger och samplade ljud – saker vi egentligen fortfarande arbetar med vid våra arbetsstationer på liknande vis. Den här datorn påverkade hela 90-talets hemdatorutveck-

ling, även om Commodore själva inte lyckades ta vara på sin framgång. Men precis som *Generation 64* kommer det här inte att bli en bok om Commodore eller om datorer, utan om människorna som använde dem för att förändra världen.

Vad verkar bli det svåraste med att skriva en Amiga-bok?

– Det svåraste med alla mina olika typer av projekt är att avgränsa dem, och den här boken är inget undantag. *Generation 64* var lättare i den meningen att Commodore 64 är en någorlunda enhetlig datormodell, den funkar nästan likadant oavsett vilken modell du äger. Amiga är ju egentligen ingen dator, utan en plattform – därav valet att skriva en bok om just modellen Amiga 500, vilket för många i dag är synonymt med ordet Amiga.

1987 släppte Commodore den dyra modellen,

"Det här blir inte en bok om datorer, utan om människorna som använde dem"

Amiga 2000, och den billigare modellen, Amiga 500. Den senare räddade hela deras ekonomi och blev en enorm framgång därför att den hade massor av grafik och ljud under huven – och blev den naturliga uppföljaren till Commodore 64, kan man säga.

Boken handlar alltså uteslutande om Amiga 500?

– Mja, nja, det är lite för tidigt att säga – men Amiga 500 är den viktiga modellen, det är först vid släppet av denna som Amiga-plattformen tar fart på riktigt. Men givetvis berör vi även andra modeller.

Vad gjorde du själv med din Amiga 500?

– Jag skaffade en Amiga 500 ganska sent, inte förrän på sensommaren 1989, har det visat sig efter att min mor hittade kvittot. Den såldes faktiskt av ingen mindre än Alexander Hofman i S.P.O.C.K. och jag vill minnas att de slog in priset fel, så jag tjänade en hel tusenlapp (*skratt*).

På det stora hela var jag nog en aspiring lamer; jag spelade väldigt mycket men var en sidekick till

lite demofolk och satsade väl främst på musiken via NoiseTracker och ProTracker. Tyvärr försvann alla mina disketter när jag lånade ut min Amiga när jag skulle plugga utomlands ett år, något jag fortfarande grämer mig åt. Det är rätt många timmar nedlagda på de där moddarna... Den enda modden som är kvar är den jag och en polare ställde upp med på The Party '92, jag var faktiskt där. Jag kallade mig då J.Wilson, tror jag, då jag inte kom på nåt tufft handle.

När Amiga 500 sjöng på sista versen orkade jag aldrig uppgradera till Workbench 2.0, utan lämnade datorlivet – det gick ju faktiskt att göra på den tiden. 1993 ryckte jag in i lumpen och inte förrän efter muck, tidigt 1995, skaffade jag min första PC, en 486 DX2-80, och på den vägen är det sedan dess.

I dag har jag ett par C64, några A600 och ett gäng rena spelkonsoler som jag tar väl hand om och räddar andras gamla disketter med och så där. Jag kan egentligen mer om Commodores datorer nu än jag kunde då, om jag ska vara ärlig. 🌐



Foto: Patrik Rosenfeld

Efterlyses: en kul dator

Varför har ingen axlat Amigans mantel?

av Jesper Odelius

Amigan var en unik maskin när den kom. Inte var den billig, men den var bra och framför allt var den något helt nytt. Det jag kan komma på som kommer i närheten är Mac, där Apple också hållit fokus på kreativitet och glädje framför tabeller och Mathematica. Men där Commodore bet i gräset klarade sig ju Apple, som vi alla vet. Tyvärr tycker jag inte riktigt att de gör jobbet.

Visst finns den lekfulla och självklara approachen hos Mac, den som också fanns hos Amigan, men jag upplever tröskeln som högre – Mac vänder sig helt enkelt inte till ungdom och än mindre till barn. För att spela *Minecraft* på datorn eller sitta i nätverksspel timme ut och timme in är inte samma sak som att på djupet interagera med en hård- och mjukvara som man gjorde med Amigan, uppfylld av en oskyldig nyfikenhet och sund teknikhunger. Det saknar jag i dag – en dator som inte är på sådant blodigt allvar som dagens maskiner.

Samtidigt är det väl ingen vågad gissning att en "strippad konsumentmaskin" skulle floppa när den ställdes mot Mac och pc – inte minst prispressade pc. Eventuella förtjänster i form av till exempel en mer intuitiv (men inte begränsad) användarupplevelse och fokus på mer lekfullhet och ett självklart skapande skulle inte motivera ett köp av vad som säkerligen också skulle klassas som en bastard med osäker framtid.

Nu kan ni inbitna Amiganer inflika och anmärka att Amiga ju lever, att det bara är att handla hårdvara, fixa mjukvara och köra. Men så upplever jag det inte alls. Tvärtom är överlevnaden av Amigan tack vare skickliga entusiaster och utvecklingen har sprungit på: Amigan är idag mångt och mycket en expertplattform, håller ni med?

Så, vad vill jag då? Jo, att Commodore vaknar ur sin törnrosasömn (*väck henne med en kyss, Jesper! red.*) och börjar trycka ut balla maskiner igen. Hårdvara med smart mjukvara som vänder sig till en publik bestående inte minst av tjejer – som inte lockas av Xbox och *Call of Duty Advanced Warfare*, som vill något mer än att spela *Barbie* på Wii eller dansa med Playstationkompisen.

Det skulle vara en mer relevant crowdsourcing-kampanj än vilken Amigaskalkampanj

som helst. Även om jag också förfärligt gärna skulle vilja ha ett nytt, fint skal till min maskin också. 🌐

Jag vill ha en dator som inte är på sådant blodigt allvar som dagens maskiner



AMIGAstore.eu

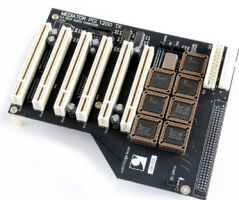
Celebrates



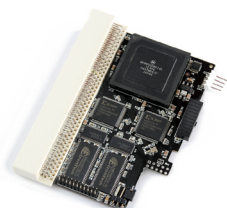
Remember when computing was fun?
It continues being fun!
Thanks to all the Amiga community



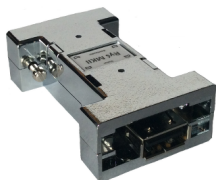
Hardware



Mediator PCI 1200 TX black
Busboard with six PCI slots for towerised Amiga 1200.
only 319,95 Euro



ACA 1221/16 turbo board
ACA 1221, the low budget turbo board for your Amiga 1200.
only 79,95 Euro



Rys MK II USB mouse adapter
The USB mouse and joystick adapter for the Amiga 1200.
only 29,95 Euro

Software



Gorky 17
The new tactic action game for AmigaOS 4.1.
only 34,95 Euro



Ask Me Up XXL
Ask Me Up XXL is a Amiga game about general knowledge.
only 29,95 Euro



Wings Battlefield
The ultimate head to head air combat game for next-gen Amigas.
only 29,95 Euro

Misc



Lemmings Heat Changing Mug
Mug with heat-sensitive imprint from the game Lemmings.
only 10,95 Euro



Commodore Patch
Must have for every Commodore fan. Dimensions 7 x 7 cm.
only 5,90 Euro



Lemmings Fridge Magnets
30 figures from the game Lemmings.
only 12,95 Euro