



Datorföreningen Updates
nyhetsblad

Nummer 3 2024

Redaktör: Bjarni Juliusson

uptime@dfupdate.se

www.dfupdate.se | update@lists.dfupdate.se | [#update](https://t.me/update) på EFnet | Svartbäcksgatan 65 i Uppsala

Färgnyckel: Föreningen Tekniskt Läsarna Nöje

I detta nummer

Nyheter	2
Stjärnuppgifter	2
Skänkes	3
Dagens dikta.py	3
Månadens dator: TI-99/4A	4
Apptajm: xdotool	7

Närmast planerade händelser

■ PR-sweatshop tisdag 28/5 kl 18-

Film, Youtube, crowdfunding, visitkort, flygblad.

■ Torsdagsverkstad torsdag 30/5 kl 18-

Lämna barnen hemma och kom och arbeta på ett av dina eftersatta projekt, eller bara visa upp det för oss andra. Verktyg, komponenter och instrument finns för elektroniskt arbete.

■ Månadens-dator-sweatshop lördag 1/6 kl 15-

Reparationsarbetet fortsätter. Månadens dator: Apple Lisa.

■ Spons-sweatshop måndag 3/6 kl 18-

Vi behöver mer spons för att finansiera Update. Hjälp till att skriva brev, ringa, och skriva PR-material.

■ Onsdagsmöte 5/6 kl 18:00

Vi träffas och äter middag. Restaurang bestäms på IRC (efnet/#update) och Updatelistan.

■ Lådsweatshop torsdag 6/6 kl 18-

Vi sorterar och krymper lådberget i L-salen.

■ Månadens-dator-sweatshop lördag 8/6 kl 15-

■ Wiki-sweatshop onsdag 12/6 kl 15-

Katalogen över samlingen behöver kompletteras.

■ Brädspelskväll onsdag 12/6 kl 18-

■ Lådsweatshop torsdag 13/6 kl 18-

■ PR-sweatshop lördag 15/6 kl 15-

■ Månadens-dator-sweatshop tisdag 18/6 kl 18-

■ Brädspelskväll onsdag 19/6 kl 18-

■ Torsdagsverkstad torsdag 20/6 kl 18-

■ Uppdatering lördag 22/6 kl 19:00

En titt på *Tcl*, ett *skriptspråk* med Bjarni Juliusson från Update. Plats: Svartbäcksgatan 65 och [via BBB](#). Se [wikin](#) för mer information.

■ Lådsweatshop måndag 24/6 kl 18-

■ Plenumsmöte onsdag 26/6 kl 18:30

Välkommen att delta i vår löpande planering och utvärdering.

■ Spons-sweatshop torsdag 27/6 kl 17-

■ Månadens-dator-sweatshop lördag 29/6 kl 15-

Alla aktiviteter äger rum i våran lokal på Svartbäcksgatan 65 om inte annat angivits.

Nyheter

■ Den 5:e maj donerade Gunnar Blomberg, alias b-gunnar, diverse datorutrustning till Update, däribland en Apple Lisa, en Exabyte-bandstation, och en TeleGuide-terminal. Tack för donationen, Gunnar, och särskilt tack för Lisan! Den är nu månadens dator i maj och juni.



Mattis Lind och Pontus Pihlgren



■ Den 18:e maj åkte Updates pontus och bjarni till Dalby Datormuseum för att hjälpa Mattis Lind måla väggarna i den nyaste delen av museet. Mattis håller på att expandera museet och har isolerat och lagt till ett nytt våningsplan i ladan.

■ Update har en tröj- och muggshop: <https://update.myspreadshop.se/>. Gör gärna reklam för den!

Stjärnuppgifter

Guldstjärnesystemet fungerar som följer: Uppgifter som letar efter någon som kan utföra dem förs upp på en tavla i L-salen. Om du tar på dig en uppgift skriver du en deadline, baserad på omfattningen. Standard är två veckor. Genomför du sedan uppgiften inom denna tid får du en guldstjärna. Misslyckas du får du därpå en vecka på dig att slutföra uppgiften utan guldstjärna. Misslyckas även detta återgår uppgiften till poolen på tavlan, och du får inte ta på dig samma uppgift igen.

Följande uppgifter finns för närvarande på tavlan:

- ★ Formge och beställa visitkort, med kontaktinfo (web + epost + besöksadress), Swish, LiberaPay, Webshop, Youtube-kanal (som behöver skapas) och eventuella andra PR-kanaler som dyker upp på PR-sweatshoppen.
- ★ Skriva pressmeddelande om utställningen och dess öppettider.
- ★ Fixa en till gatupratat, till andra sidan gatan, gärna större. Alternativt två, så kan vi ersätta den pyttiga av trä vi har nu.
- ★ Öppetskyltar till dörr/fönster.
- ★ Kontakta medlemmar som glömt att betala medlemsavgiften.
- ★ Wiki Cray Y-MP EL (Riemann); datorn saknar wiki-sida.
- ★ Donera en pappersgiljotin.
- ★ Handbok Macintosh; behöver skrivas till utställningen.
- ★ Handbok Amiga; till utställningen.

Det gäller också som vanligt att tecknande av ett nytt sponsoravtal alltid ger en guldstjärna. Dessutom ger det från och med nu en stjärna att spela in en film att publicera på webben, vilket vi borde göra så ofta som möjligt. Det ger nu också en stjärna att hålla en uppdatering.

Skänkes

↓ Bjarni skänker en Data General Eclipse S/280 samt en Dasher-terminal.



Update skänker en TeleGuide-terminal samt en utgallrad LA180-manual. ↓



Är du sugen på något av föremålen, skriv till uptime@dfupdate.se.

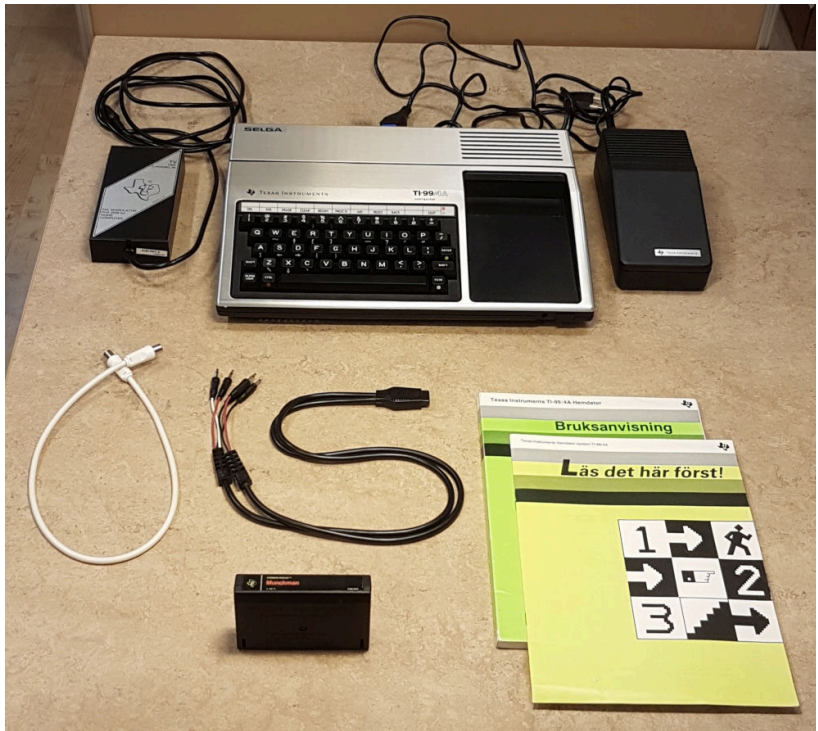


Dagens dikta.py

Jersey-hombren Sean serverar även emiren Herr Ä en öl i Ljusne.
En öl till Dr. Janne, fena på AI och lycra-dåre. Påven? En öl.
Heureka! Tänk - jag anlägger LCHF-pagoden "Fenris-ejderen" i Århus!
Byggtöcken! Husreverier! Hans neuroner ser digital-TV-antennerna i MDF,
rosa skåp med 2000-punds Dior-fanér, en tapet, en komet med råa ägg på,
endast valfett i konditoriet! Kom med i en ny diet-era!

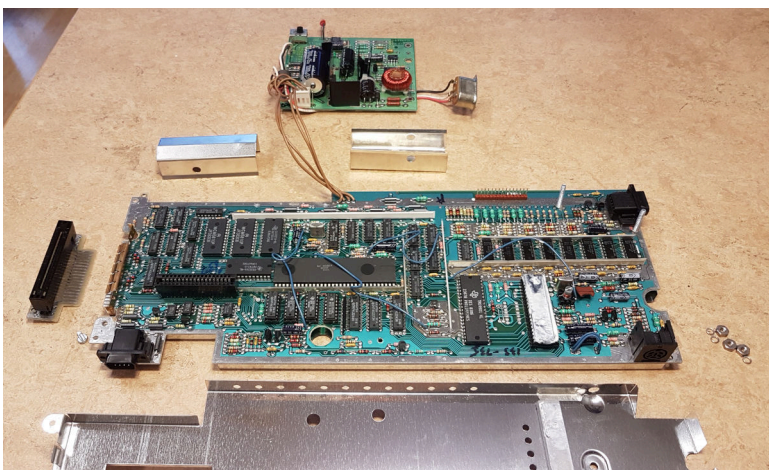
Månadens dator: TI-99/4A

I april tog vi fram Updates nyligen donerade TI-99/4A, en datormodell som introducerades 1981 av det ej för persondatorer kända företaget Texas Instruments som den första 16-bitars hemdatorn. Modellen är till det yttre ganska typisk för sin tid: hela datorn är inbyggd i tangentbordet, har kontakter för kassettbandspelare, joysticks, programkassetter och en TV som fungerar som bildskärm, och matas av en extern transformator som kopplas in i väggen. I bilden till höger syns datorn, transformatorn (t.h.), den separata och enorma RF-modulaturen för TV-signalen (t.v.), TV-antennsladden, sladden för anslutning av (upp till två) kassettbandspelare (mitten), bruksanvisning, samt en programkassett med spelet Munchman.



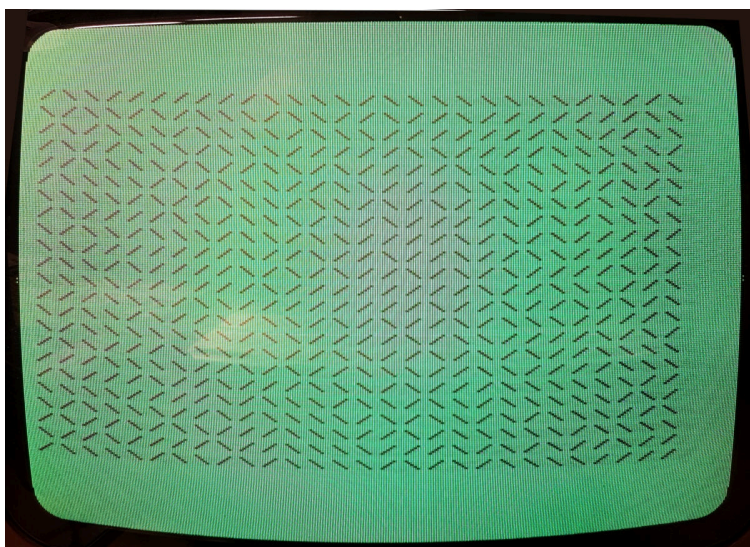
Kablaget till Updates exemplar hade smält frigoliten den var packad i, som följde med som ett vitt ormskinn när man lyfte ur datorn och tillbehören ur förpackningen. Lyckligtvis var det ganska lätt att dra av ormskinnet.

Vi mätte sedan spänningen från den externa transformatorn för att försäkra oss om att vi inte skulle skada datorn när vi kopplade in den, och så öppnade vi datorn och undersökte den för att se att ingenting var svart, rostigt, smutsigt etc. Men allt såg bara fint ut.



Två kretskort finns inuti datorn: det ena är själva datorn och det andra, mindre, är nätdelen. Det stora kretskortet täcks nästan helt av en metallsköld. Denna agerar också kylfläns för ett av chippen. Vi passade på att byta kylpastan när vi ändå tog av skölden. Lustigt nog sitter kontakten för programkassetten själv på ett litet kretskort istället i en motsvarande kontakt på själva datorkretskortet, för att vända den 90°. Var den tänkt att vara vertikal från början? Det är ju vanligare.

Hur som helst var allt i sin ordning med datorn, så vi kopplade in den i en TV och slog igång den, och den visade sin startskärm. På TI-99/4A hamnar man nämligen inte direkt i BASIC, utan man kommer först till en startskärm och en meny där man kan välja att köra ett program från ROM-kassett om en sådan är istucken. Som test skrev vi en version av "10 PRINT", ett BASIC-program som slumpmässigt skriver ut \- och /-tecken, vilket ger en effekt som ser ut som en labyrint. På TI-99/4A fungerar detta dock inte så bra, eftersom textläget har väldigt stora glapp mellan tecknen på skärmen:



Dessutom går det i *slow motion* tack vare att PRINT-kommandot i BASIC-dialekten på TI-99/4A inte har stöd för att avsluta utskriften med ';' för att förhindra radbrytning. Således måste man bygga ihop en sträng för varje rad som ska skrivas ut, vilket går långsamt. Dessutom är själva BASIC:en i sig långsam, och datorn är också långsam (mer om det strax). Så det tog ungefär två sekunder per rad att skriva ut labyrinten. På en Commodore 64 ger den trivialaste implementationen ungefär en och en halv rad per sekund, som jämförelse. Synd att vi inte har några andra programmeringsverktyg förutom den obra inbyggda BASIC:en.

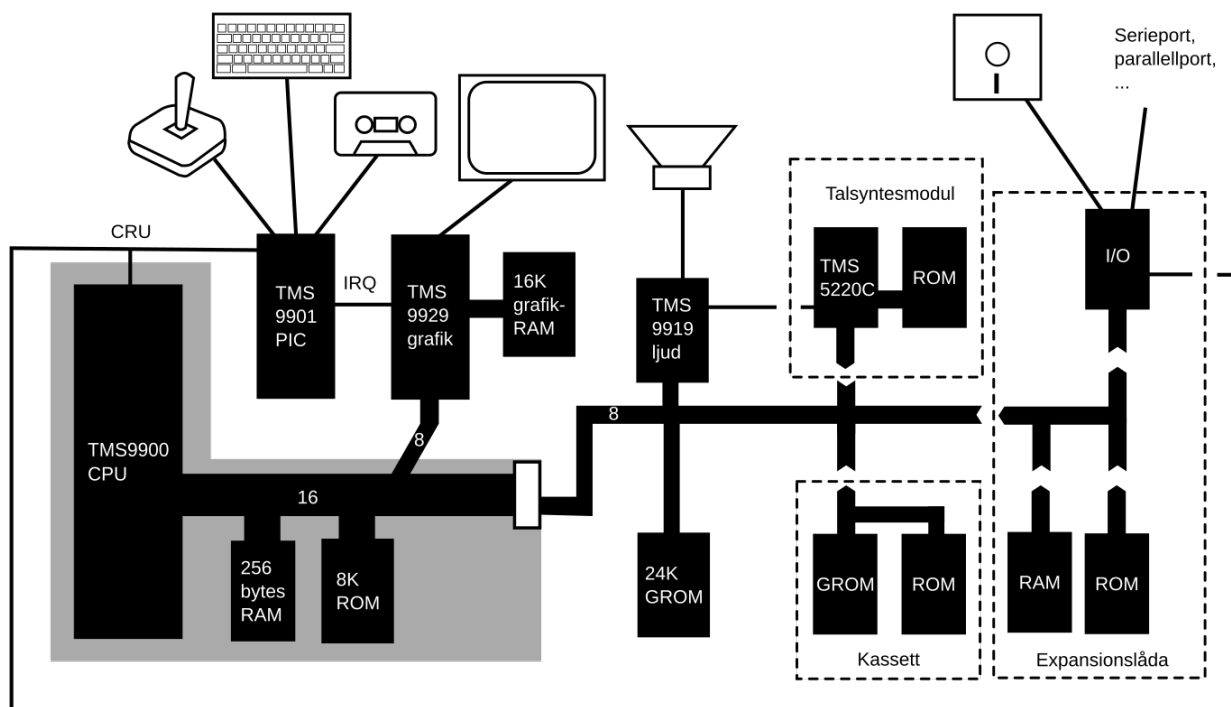
Sen spelade vi lite Munchman. Det är en halvdan kopia på Pacman. Inte jättekul utan joystick. Förutom detta spel har Update just nu ingen mjukvara för TI-99/4A, så mer än så testade vi den inte, men datorn verkar vara i utmärkt fungerande skick. Vi hittade senare ett arkiv med band-images som vi skulle kunna prova att ladda från en laptop via kassettinterfejset, men det hann vi inte med den här gången.



Kanske finns lite programmeringsverktyg att prova. En assembler skulle vara särskilt intressant att hitta, eftersom TI-99/4A har en mycket udda CPU: TI:s egna 16-bitars TMS9900 från 1976, som förutom i TI-99/4 och /4A mest användes i de mindre modellerna i Texas Instruments egna minidatorfamilj TI-990. Valet av processor var förmodligen en del av anledningen till den brist på tredjepartsmjukvara som bidrog till att TI-99/4A lades ner efter tre år på marknaden, under vilka priset sänkts i omgångar från 525 USD (20.000kr idag) ner till 49 USD, i konkurrens med Commodore, med brakförluster för TI p.g.a höga tillverkningskostnader. Brist på dokumentation och fientlighet mot tredjepartsutvecklare bidrog också, liksom den udda BASIC-dialekten.

Nu ska vi prata lite om varför TI-99/4A är så långsam!

Ta en titt på diagrammet här nedan. Det beskriver arkitekturen i datorn. Varje svart ruta är ett chip (eller grupp av RAM/ROM-chip), och dessa är ihopkopplade via bussar. Överst syns de gränssnitt som datorn tillhandahåller för kommunikation med omvärlden: tangentbord, kassettband, bildskärm, etc. De streckade rutorna är tillsatser man kunde köpa. Expansionslådan är en stor låda, större än datorn, som pluggas in via en sladd på datorns högra sida, och i vilken man sedan kan stoppa expansionskort. Det fanns också expansionsmoduler som pluggades in direkt i sidan av datorn, och sedan i varandra i en allt längre human centipede ju rikare föräldrar man hade.



Längst till vänster finns processorn, TMS9900, som har en 16 bitars databuss. Allting som är i den gråa rutan är kopplat till den bussen. Som ni ser är det inte så mycket: 8Kbyte ROM och 256 bytes RAM. Från 16-bitarsbussen går en liten 8-bitars gren till grafikkretsen, som bara använder ena halvan av bussen. Därifrån går en separat 8-bitars buss till grafikminnet. Det är dessa 16K som i praktiken utgör det huvudsakliga arbetsminnet, men det är inte direkt åtkomligt för processorn. Istället måste man skriva en önskad adress i ett pekarregister i grafikchippet och sedan läsa ut data ur grafikminnet via grafikchippet 8 bitar åt gången. Där har vi anledning nr 1 till att saker och ting går så långsamt.

Anledning nr 2 ser vi i den lilla vita rutan som kopplar samman 16-bitarsbussen med en 8-bitars buss som det mesta annat sitter på. Detta är ett multiplexergränssnitt som för varje 16-bitars buss-operation stoppar processorn och gör två separata operationer på 8-bitarsbussen. Detta är i stor utsträckning rent slöseri med tid, för både ljudchippet och GROM:et är hur som helst bara mappade på de respektive jämna byte-adresserna, så den ena av de två busscyklerna överför inga data.

Anledning 3 är att GROM-chippen inte har någon adressbuss utan man skriver istället adressen en byte i taget till ett register i chippet via databussen. Sekvensiella läsningar går sedan på en cykel (som ett normalt ROM), men ickesekvensiella läsningar tar alltså tre cykler, plus overhead. Processorn gör dessutom alltid en läsning före varje skrivning, så det är en extra bortkastad busscykel. Dessutom är GROM-chippen långsammare än processorn och stoppar in extra wait states.

Anledning 4 är att det såklart inte går att ha exekverbar maskinkod i GROM, eftersom det inte är direkt adresserat av processorn. Så istället innehåller ROM:et i den gråa rutan en virtuell maskin vid namn GPL, och så är BASIC-tolken skriven i GPL-bytekod. Så när du kör ditt BASIC-program är det GPL-tolken som kör BASIC-tolken, som den läser från stenlångsamt GROM, och som i sin tur läser ditt program genom grafikchippet från skärminnet. Nästan ingenting är 16 bitar brett.

Anledning 5 är att CPU:n inte har några register, utan alla operationer är minne-minne. Dessutom är den inte ens särskilt snabb; klockan går i 3MHz, och de flesta instruktionerna tar omkring 14 cykler.

"CRU" i diagrammet är för övrigt en I/O-buss som är seriell, och alltså inte heller så värst snabb.

Apptajm: xdotool

Ibland händer det att man vill skriptat händelser i X11, t.ex. tangenttryckningar, flyttande eller ändring av storlek av fönster, musklick, etc. Kanske vill man exempelvis binda en tangentkombination i sin fönsterhanterare till att göra nånting som fönsterhanteraren i sig inte har stöd för. Ett fall som uppstod i T-salen var att binda volymtangenterna på X-terminalerna till att höja och sänka volymen på kakrafoon, vars volymkontroll man kom åt via en telnet-session. Man hade ett terminalfönster uppe med volymkontrollen i, men hur får man volymknapparna att skicka rätt input till detta fönster?

I den sortens situationer har man nytta för xdotool. Detta är ett mycket enkelt och lättanvänt program, som på sin kommandorad tar en lista av kommandon för att välja ut fönster och göra saker med dem. Ett typiskt bruk kan se ut såhär:

```
$ xdotool search --name foo windowactivate
```

Invokeringen ovan anger två kommandon: search (med parametrarna --name foo) och windowactivate. Det första av dessa söker upp ett fönster med namnet "foo", och det senare raisar och fokuserar detta fönster.

Om man bara kör ett search-kommando (eller om man kör det sist av alla kommandon) skriver xdotool ut resultatet:

```
$ xdotool search --name foo
184549388
```

Numret som skrivs ut är fönster-ID:t för det matchande fönstret. Om flera fönster matchar skrivs alla deras ID:n ut, ett per rad. Om inget matchar avslutar sig xdotool med returstatus 1. På så vis kan t.ex. ett shellskript kolla om ett fönster finns genom att titta på returvärdet:

```
$ if xdotool search --name xterm >/dev/null ; then echo Fönstret finns; fi
Fönstret finns
```

Kommandon som manipulerar fönster behöver veta vilket fönster de ska manipulera. En del av dessa tar ett fönster-ID med en switch, --window, och en del tar det utan switchen. windowactivate t.ex. tar fönster-ID:t utan någon switch:

```
$ xdotool windowactivate 184549388
```

Men man kan också använda en sökning efter fönster för att hitta ID-numren automatiskt, och använda dem i ett efterföljande kommando. I detta fall sätter search-kommandot en "fönsterstack" innehållande ett eller flera fönster-ID:n (i den ordning fönstren är stackade på skärmen), och efterföljande kommandon kan referera till fönster i stacken med %n, där n är indexet i stacken. Så för att söka efter fönster som heter "foo" och sedan aktivera det första som hittas kan vi köra följande:

```
$ xdotool search --name foo windowactivate %1
```

Vanligtvis är dock %1 default, och kan utelämnas. Det var på det sättet som det första exemplet i den här artikeln fungerade. Man kan också ange %@ för att köra kommandot på alla fönstren i fönsterstacken.

Förutom att söka med search kan man också välja det fönster som har fokus just nu med getwindowfocus eller låta användaren välja ett fönster genom att klicka på det, med selectwindow. Dessa sätter fönsterstacken på samma sätt som search, fast alltid med bara ett fönster. search kan förresten söka på annat än namnet — se mansidan för xdotool.

Andra saker man kan göra med fönster är t.ex. att skicka tangenttryckningar och musklick till dem, ändra fönsterstorleken, flytta fönster, stänga fönster, eller försöka slå ihjäl klienten som äger fönstret. Har man ett GUI-program som har hängt sig kan följande alltså vara användbart:

```
$ xdotool selectwindow windowkill
```

Tangenttryckningar kan skickas med kommandot key, som begriper namn på tangenter och även tangentkombinationer:

```
$ xdotool selectwindow key Ctrl+D
```

Textsträngar skickas enklast med kommandot type:

```
$ xdotool selectwindow type "Update gör dig vacker!"
```

För att flytta och ändra storleken på fönster finns följande kommandon:

```
$ xdotool selectwindow windowmove 0 0 windowsize 50% 100%
```

Här ser vi att vi kan köra mer än ett kommando efter varann på samma fönsterstack. Vi ser också att windowmove tar två parametrar, som såklart är X- och Y-koordinater, att windowsize också tar två parametrar, bredd och höjd, och att det kommandot förstår procentsatser, vilket gör att man t.ex. kan resiza ett fönster till att ta upp halva skärmen, som i det här exemplet.

Förutom att göra saker med fönster kan vi också göra saker med musen. För att teleportera muspekaren till en given plats på skärmen gör vi som följer:

```
$ xdotool mousemove 0 0
```

Vi kan också flytta musen relativt till där den är:

```
$ xdotool mousemove_relative 10 10
```

Och så kan vi klicka med de olika knapparna:

```
$ xdotool click 1 click 2 click 3
```

Man kan också ange ett filnamn istället för kommandon på kommandoraden för att be xdotool att läsa kommandon från en fil. Om man anger ett "-" läser programmet från stdin. Använder man en shebang-rad kan man göra ett körbart xdotool-skript. Detta kan man sedan köra med kommandoradsparametrar, som man i skriptet kan referera till med \$n, precis som i ett shellskript:

```
$ cat säg_nånting.xdt
#!/usr/bin/env xdotool
search --name $1 windowfocus
type "$2"
key Return
$ ./säg_nånting.xdt irssi "Update gör dig vacker!"
```

xdotool är alltså ett till sånt där enkelt program som man sällan har användning för, men när man har det är man väldigt glad att det finns. För mer information, se, som vanligt, mansidan.

Tack från Uptime för den här gången! Har du artiklar, insändare, nyheter, eller utrustning du vill skänka eller byta? Skriv till uptime@dfupdate.se! Vi ses om en månad. o/