

Datorteknik RT3 ht 2008

Johan Arvelius

26 november 2008

Det här dokumentet i senaste version:

HTML <http://www.irf.se/~johan/datorteknik/>

PDF <http://www.irf.se/~johan/datorteknik/planering.pdf>

Introduktion

Vi har tillgång till en halv klassuppsättning datorer, och kurstiden medger i stort sett bara helklassundervisning. Ni kommer därför att delas in i par där varje par skall få ta hand om en dator. Min vision med den här kursen är att den skall bli en sammanhållen del som kretsar kring denna dator. Vi börjar från början med att plocka isär den och se på innehållet, fortsätter med att installera operativsystem och sedan tillämpningar.

Min plan idag, 22:a sept, är att vi skall jobba i **Xubuntu GNU/Linux** och att tillämpningar främst kretsar kring att sätta upp webserver på den och att programmeringen ska ske i **PHP**. Om det finns goda argument för andra OS, tillämpningar eller programmeringsspråk är jag öppen för sådana förändringar förutsatt att jag klarar av dem.

Vi har en lärobok Eriksson som vi kommer att använda framförallt i början, sedan blir det mest hänvisningar till materiel på nätet och en del egen informationssökning på nätet.

Det mesta arbetet tänker jag mig sker i labbparen men i slutänden tänker jag ha en examination som till stor del ska vara egenhändigt arbete vid den dator ni har haft under kursen, det blir då av naturen i halvklass.

Läsanvisningar

Det här dokumentet är det tänkt skall växa fram under kursens gång så att det innan varje kurstillfälle ska finnas detaljerade anvisningar för det tillfället. Läsanvisningar utan särskild markering, övningar med nummer som Ö1 etc. och frågor med nummer som Q1 etc är det tänkt att alla ska göra. Moment markerade med • ska redovisas parvis i labbparen. Moment markerade med * är extrauppgifter för de som hinner med, fler stjärnor är högre grad av överkurskaraktär. I grunden är varje rubrik en arbetsdag.

1 Hårdvara

- Mål** Målet med dagens övningar är att vi skall lära oss vad de olika delarna i datorn har för uppgift, hur de ser ut och hur man kan byta ut dem. Dessutom skall vi få ett grepp om vilka standarder som används idag för olika delar och lite om vad som är normal prestanda för datorer idag.
- Läs** sidorna 17–24, 31–42 i Eriksson.
- Övningar**
- Ö1 Övning 7 sid 54 i Eriksson.
 - Ö2 Övning 9 ibid.
 - Ö3 Montera isär datorn. **Ta det försiktigt med elektriska urladdningar, ta alltid först i chassikonstruktionen för att jorda dig.** Montera ur: floppyenheten, CD-enhet, hårddisk(ar) och minneskretsar.
 - Ö4 Identifiera processorn eller processorkylaren om inte processorn är synlig. Kontrollera i samråd med läraren om det går att ta ur processorn utan att lossa kylflänsen från den.
 - *Ö5 Om det går, ta ur processorn. **Ta inte isär processorn och kylflänsen.**
 - *Ö6 Om ni vill kan ni ta ur moderkortet också.
 - Ö7 Skriv av tillämpliga delar av tabellen i Övning 6 sid 30 i Eriksson och fyll i de delar du kan.
 - *Ö8 Se om ni kan hitta chipet med Basic Input/Output System (BIOS) på moderkortet. Sitter det i en sockel där det går att byta ut?
 - *Ö9 Sök på nätet om ni kan hitta en bild eller manual till moderkortet som hjälper till att identifiera fler av delarna ni ser på kortet.
 - Ö10 Montera tillbaks alla lösa delar i lådan igen.
- Besvara**
- Q1 Vilka formfaktorer är det som gäller för nya datorer idag? Vilka olika tillämpningar används de i och vilka prestanda ger de? En startpunkt skulle kunna vara <http://www.tomshardware.se/>.
 - Q2 På sid 34–35 omnämns processorsocklar av slottyp, används de idag? Finns de i våra kursdatorer? Vilka socklar är vanliga idag? Hur många anslutningspinnar har de ungefär?
- Redovisa**
- Sammanställ svaren på övning Ö7, och frågorna Q1 och Q2 och skicka in dem per epost till Johan.
 - * Att skicka in i ett öppet format som Postscript (PS), Portable Document Format (PDF), eXensible Markup Language (XML) eller Hypertext Markup Language (HTML) är plus.
 - ** Att skicka in svaret typsatt i ett typsättningsprogram (t.ex. L^AT_EX) istället för att skriva i en ordbehandlare är överkurs. En bra svensk manual finns på <http://www.dcg.lth.se/perf/handledning/>.

2 BIOS, bussar, minnen, diskar

- Mål** *Veckans* övningar går ut på att vi skall bli bekanta med hur datorn styrs på den lägsta mjukvarunivån BIOS och hur den interna kommunikationen i datorn är uppbyggd och hur datorn håller reda på alla data i olika typer av minneskretsar. Vi ska gå också gå igenom hur datorn lagrar data på hårddiskar, och vad som menas med formatering, partitionering och Redundant Array of Independent Discs (RAID).
- Läs** Eriksson, sidorna: 42(BIOS)–51, 76–83(PCMCIA), 89(TP)–91, 94–107(DDR DRAM), 113–115, 122–128(Diskcache), 132–139 och nedanstående om processorer.
- Övning** Koppla in tangentbord och skärm till datorn, stoppa i sladden och tryck på strömbrytaren. Se till att du kan komma till BIOS-konfigurationen. Ta reda på hur du manövrerar dig fram i konfigurationen.
- Ö11 Gå igenom BIOS-konfigurationen. Fyll i de saker som saknas i tabellen du gjorde i övning Ö7.
 - Ö12 Vi ska installera operativsystem från Compact Disc (CD). Tänk igenom hur i vilken ordning BIOS ska kolla efter enheter att starta från. Gör dessa inställningar i BIOS.
 - Ö13 Om kontakter och verktyg för byte av TP-kontakter finns tillgängligt ska ni kontaktera en egen TP-kabel. Följ instruktioner på http://www.kjell.com/?page=/kjellfakta/tp_kontaktering/.
- Besvara** Eftersom större delen av dagen går till besök på IRF kommer vi att diskutera frågorna nedan på fredag 10:e oktober.
- Q3 Vilka olika typer av minnen sitter normalt i en bordsdator? Rangordna dem efter hur snabba de är. Vad är normala storlekar för de olika typerna, vad är deras uppgifter?
- Q4 Vilka bussar är det som gäller idag? Gå ut på nätet och titta på moderkort, grafikkort och annan utrustning.
- *Q5 Universal Serial Bus (USB) och firewire (Institute of Electrical and Electronics Engineers (IEEE)-1394) är två olika moderna sätt att ansluta externa saker till datorn, de kan verka lika men är i grunden olika koncept ta reda på hur? USB 2 har den nominella hastigheten 480 MB/s medans firewire-400 har den nominella hastigheten 400 MB/s trots det ger det ofta snabbare överföring att ansluta saker via fw-400, varför? Sök svaren på nätet.
- Q6 Om man har två hårddiskar om 150 GB respektive 200 GB i övrigt lika och vill ha maximal hastighet och största möjliga utrymme, hur ska man då partitionera dem och koppla dem i RAID?
- Q7 Om man istället vill använda ovanstående diskar för att spara användarnas data så säkert som möjligt, hur ska man då partitionera dem och koppla dem i RAID?
- Q8 Vad finns det för anledningar till att partitionera en hårddisk?

2.1 Processorer

Datorns Central Processing Unit (CPU), eller processor är idag en mycket invecklad skapelse. Det finns många faktorer som avgör hur snabb den är. Den del av processorn som har avkodare och beräkningskretsar och gör själva beräkningsjobbet kallas kärna. Ej

att förväxla med operativsystemets kärna. Hur de är uppbyggda, hur många steg datat bearbetas vid sin passage genom kärnan är avgörande för hur snabbt den arbetar. Olika arkitekturer är i regel bra på olika typer av uppgifter, och det är inte ovanligt med 'tester' från tillverkare eller fanatiska användare med speciellt tillrättalagda arbeten som visar att just den processorn är mycket snabbare än konkurrenterna.

2.1.1 Bussbredd

Kärnan arbetar genom att ta in ett visst antal databitar i taget för bearbetning, detta är den interna bussbredden. Nya datorer som säljs idag har 64 bitars bussbredd, medans datorerna i den generation vi har på kursen har 32 bitar. Bussbredderna för konsumentdatorer var från början 8 bitar och har sedan gått via 16 och 32 bitar till att de senaste åren domineras av 64 bitarprocessorer.

En bidragande orsak förutom att processorerna med bredare bussar kan bearbeta mer data per klockcykel (se nedan) är att storleken på internminne blir över 4 Gigabyte (GB). 32 bitar kan nämligen användas till att skriva tal upp till ca 4000000000, och man måste sedan använda antingen två sådan tal för större minne eller något annat knep men det blir aldrig lika effektivt som att ha en bussbredd som direkt kan adressera hela minnet.

Hur mycket data processorn matas med åt gången bestäms av programmen som körs. Det måste bestämmas vid kompileringen av programmet. American Micro Devices (AMD) satsade stort på att öka bussbredden för några år sedan och sålde många 64-bitars processorer, det dröjde dock länge tills Microsoft (MS) Windows kom i en 64-bitarsversion så i praktiken var det bara linuxanvändare som hade nytta av den större bussbredden under tiden. Ännu idag är det många kommersiella tillverkare som inte levererar drivrutiner kompillerade för 64-bitars bussbredd så att det fortfarande är vanligt att ha 32-bitars operativsystem för att kunna använda all kringutrustning.

2.1.2 Klockfrekvens

Bussbredden bestämmer alltså hur stora tuggor data kärnan kan ta på en gång och hur ofta den kan ta en ny tugga bestäms av klockfrekvensen som helt enkelt är ett uttryck för hur många kodsnuttar om vardera bussbredden som kärnan kan ta emot varje sekund.

2.1.3 Minnen

Det stora primärminne som sitter i lösa kretsar på moderkortet är mycket långsammare på att svara än en klockcykel i processorn därför använder man två minnesbuffertar som sitter inne i själva processorhöljet dit man kopierar alla saker som processorn efterfrågar från primärminnet samtidigt som det går till kärnan. Det gör man för att det är vanligt att processorn efterfrågar samma information igen snabbt efter och då finns den snabbare till hands. De data som kommer ut från kärnan går också först till dessa buffertminnen och sedan vidare till primärminnet på moderkortet. Det minsta och snabbaste minnet som sitter närmast kärnan kallas L1-cache och det större minnet som har lite längre accesstid kallas L2-cache. Moderna och invecklade processorer som Intel Core försöker gissa i förväg vad processorn kommer att efterfråga och laddar över det till cacheminna.

2.1.4 Flera kärnor

Genom att kapsla in två likadana kärnor brevid varandra i samma processor blir det som att ha två lika processorer som jobbar parallellt. Det har på sistone börjat slå igenom på vanliga konsumentdatorer.

Problemet är att ett normalt datorprogram måste räknas igenom "från början till slut" d.v.s. delberäkningarna måste komma efter varandra i tur och ordning. De två kärnorna kan då inte samarbeta med samma problem. Till stor del fungerar det så att de två kärnorna helt enkelt delar upp arbetet så att de håller på med varsitt program. För

att utnyttja kapaciteten bättre försöker man skriva program som delas upp i separata sk. trådar som kan köras parallellt i två olika kärnor. Det är betydligt svårare och ställer krav på programmeraren. Problemet har dock varit behandlat länge eftersom större datorer ofta har flera processorer som ska dela på jobbet.

Kärnorna i flerprocessorkärnor har alltid varsin L1-cache men beroende på arkitekturen kan de ha varsin eller dela på L2-cache. Om de delar på den måste var och en adressera sin del så de kan inte använda varandras cacheade data, men fördelningen kan vara olika om de har olika krävande jobb.

2.1.5 Dagsläget

Avsnittet om processorer är hopplöst utdaterat i boken. Kortfattat är det som hänt sedan boken skrevs följande: Vid millenieskiftet dominerade Intel Pentium III och AMD Athlon marknaden. Intel valde en väg med en relativt enkel arkitektur på processor som genom att vara enkel skulle gå att köra med väldigt hög klockfrekvens och därigenom bli snabbare – Pentium 4. AMD valde att satsa på mer på att göra en mer invecklad processor som skulle kunna konkurrera om att göra samma mängd beräkningar på en lägre klockfrekvens.

Intels P4 blev med högre frekvens att dra ganska mycket ström och därmed varm och krävde mycket kylning. Därför började man efter ett tag en parallell utveckling av en ny gren processorer för bärbara datorer – Pentium M med intelligentare design. Intel var något tidigare med dubbla kärnor på konsumentmarknaden, medans AMD var snabbare att gå över till 64 bitars bussbredd. De vanligaste processorerna från båda tillverkarna har nu både 64-bitarsprocessorer och dubbla kärnor. Till snabbare datorer finns varianter med fler kärnor per processor.

3 Installation

Mål Dagens mål är att alla ska ha ett fungerande operativsystem på sina datorer. Och att veckans mål enligt tidigare ska uppnås.

Övning Installera **Xubuntu GNU/Linux** på datorerna. Generellt verkar det största problemet vara partitioneraren vid installationen. Den ska dela upp hårddisken i delar s. k. partitioner. Eftersom det är ödesdigert om den gör fel och data på disken kan bli oläsligt är den väldigt kinkig med att kolla att det går att skriva och läsa felfritt från disken. Om det installationssteget hänger sig är det lättast att börja om och hoppas. Flertalet av datorerna har klarat en provinstallation nu i höst även om det ibland har varit efter en del knorr från partitioneraren.

Ö14 Starta datorn och lägg direkt i installations-CDn, om ni lyckades med att sätta bootordning i övning **Ö12** så ska den sätta igång och installera. Följ instruktioner på skärmen.

Dagens övning är att installera operativsystem på datorerna. Jag har valt Gnu's Not Unix (GNU)/Linux-distributionen Xubuntu, en slimmad version av Ubuntu till den här kursen. Det är derivat av Debian som är en av de äldsta och största distributionerna av linux. Debian är väldigt mån om att ha välfungerande stabila system och lägger väldigt lång tid på att avbugga system. De är därför lite lätt ålderstigna redan när de kommer ut. De är populära att använda för servrar där stabilitet är högre prioriterat än att ha senaste versioner av program. Hemanvändare som kör Debian brukar använda testing eller unstable som uttestningsversioner där nya programversioner provas för att ha modernare program. Ubuntu tar i princip en ögonblicksbild av Debian unstable och filar på den, jagar efter fel (buggar) i programmen och ser till att slipa till användargränssnitt så att systemet ska bli lättanvänt och släpper en distribution efter sex månader och börjar så om med en ny ögonblicksbild av Debian unstable.

Systemet är uppbyggt kring kärnan Linux, som är en sköter om de mest grundläggande instruktionerna i datorn alltså styra det mesta av hårdvaran. Kärnan styr också vilka andra program som får starta och hur de får dela på de resurser som finns. Linuxkärnan är från början skriven av finländaren Linus Torvalds men underhålls idag av många frivilliga under hans ledning. Källkoden till linux är släppt under General Public License (GPL).

Den grundläggande funktionaliteten för att det ska bli ett användbart operativsystem är en uppsjö med samverkande program i projektet GNU.

Linux och GNU är sen hoppackat med en mängd andra användbara program i en distribution. Distributionens roll är att plocka ihop programversioner som fungerar tillsammans, hålla reda på att systemet sätts upp på ett strukturerat sätt och göra det enkelt för den som administrerar datorn att installera nya program. I vårt fall är alltså distributionen Xubuntu, som är en variant på Ubuntu, som i sin tur är baserad på distributionen Debian.

Debian och därmed även Ubuntu har ett egenutvecklat pakethanteringssystem där program är hopbuntade i paket. I paketen finns förutom själva programmen information om vilka andra paket som behövs för att det ska fungera, hela systemet hålls ihop av en pakethanterare som läser denna information och automatiskt installerar dem.

4 Linux, GNU och X

- Mål** Efter dagens övningar ska alla kunna
1. söka hjälp i operativsystemet,
 2. leta rätt på och installera program som ingår i distributionen och lösa progampaket,
 3. kunna leta reda på filer och leta i innehållet i filer,
 4. förstå systemet med läs och skrivrättigheter och kunna ändra ägare och rättigheter till filer,
 5. lägga till användare,
 6. kunna starta X-sessioner och kunna byta mellan textkonsoller och X-sessioner,
 7. kontrollera systemets pågående processer och kunna stänga ner processer som hängt sig,
 8. kunna redigera och läsa textfiler i texteditor och via kommandon i textkonsoll och
 9. ha ett installerat system som har de verktyg som behövs för ovanstående.
- Läs**
- Stycke 1.3–1.4 i Andersson <http://www.sslug.dk/artikler/gnulinux/node6.html>
 - Nedanstående.
 - Walleij sid 84–96, (utkopierad).
- Övning**
- Ö15 Gör en testfil med lite text genom att använda texteditorn mousepad. Spara den i din hemkatalog.
- Ö16 Det finns två sätt att få fram en textterminal. Prova dem:
1. Tryck samtidigt ner knapparna `Alt-Ctrl-F1` på tangentbordet. Logga in vid prompten. Hoppa tillbaks till det grafiska systemet genom att trycka `Alt-Ctrl-F7`.
 2. Leta reda på en textterminal bland programmen i menysystemet.
- Ö17 Gör övningarna 2, 3, 9–13 (onummerade på websidan) i Andersson 1.5 <http://www.sslug.dk/artikler/gnulinux/node8.html>
- Ö18 Läs igenom stycket 3 i Andersson <http://www.sslug.dk/artikler/gnulinux/node16.html> och häng med genom att prova de kommandon han går igenom. Gör lite egna textfiler i texteditorn mousepad och spara till hemkatalogen som ni kan använda för att testa kommandona.
- Ö19 Skapa en ny användare så att ni har varsin. Det kan finnas ett grafiskt interface för att göra det, se om ni kan hitta. Annars är kommandot `adduser`, använd hjälpen för att ta reda på hur det fungerar.
- Ö20 Använd tangentkombinationen `Alt-Ctrl-F1` för att få upp en textkonsoll. Logga in med den andra användaren och starta en X-session för den användaren parallellt med den första med `startx - :1` eller `xinit - :1`. Orkar datorn med det? Gå tillbaks till de två X-sessionerna med `Alt-Ctrl-F7` respektive `Alt-Ctrl-F9` och se att de fungerar samtidigt. Vad händer när man stänger ner den andra sessionen? Vad händer när man stänger ner den första? Gör motsvarande operation genom att klicka på “avsluta” och välja “växla användare”.

- Ö21 Öppna pakethanteraren synaptic och använd sökfunktionen för att söka efter emacs. Då får ni upp en lång lista på programpaket som de allra flesta insticksmoduler för editorn emacs. Själva editorn finns i två implementationer, Gnu EMACS, (paketet som heter emacs), och XEmacs siffrorna efter namnet betyder bara vilket versionsnummer det är. Markera antingen emacs eller xemacs och tryck på knappen verkställ. Godkänn installationen av de paket som föreslås.
- Ö22 Använd växeln `-k` till man för att söka reda på vilka texteditorer som finns installerade. Är det några som har mystiskt lika beskrivningar? Vad beror det på? Går det att hitta på editorerna i menysystemet?
- Ö23 Öppna någon av textfilerna ni gjort och se om någon av editorerna verkar bra. Prova och se att ni klarar att ändra och spara med åtminstone en editor som körs i textmod d. v. s. i textfönstret med kommandoraden.
- *Ö24 Försök att bemästra vi.
- Ö25 Öppna en webbsida med firefox. Spara ner källkoden till den genom att högerklicka och välj "spara sida som" i menyn. Öppna den i texteditorerna ni provat och se om det finns någon specialmod för HTML. Ger den syntax-highlight och automatisk indentering?
- Ö26 Testa ordbehandlaren Abiword. Vilka olika format kan ordbehandlaren läsa och skriva? Sök rätt på ett dokument i MS word-format och se om ni kan öppna det med ordbehandlaren. Prova om det går att spara i PDF-format. Öppna och kontrollera resultatet med PDF-läsaren.
- Ö27 Se till att ni kan komma åt er e-post från datorn. Om inte det går att fixa, skaffa ett konto på någon gratistjänst med webbgränssnitt.
- Ö28 Lista *alla* filer för de båda användarna ni har på systemet d. v. s. även sk. dolda filer på ett sätt som visar vilka rättigheter de har antingen med ett textkommando eller i en filhanterare. Spara resultatet på ett vis som ni kan skicka med i redovisningsfilen till mig.
- *Ö29 Läs stycke 1.2 i Andersson <http://www.sslug.dk/artikler/gnulinux/node5.html> och gör resten av övningarna i 1.5.
- *Ö30 Ta reda på hur många länkar det finns i `/etc/alternatives` hur många av dem går till filer som har ändelsen `.gz`?

4.1 Hjälp

Det klassiska hjälpkommandot i UNIX-system är `man`. Om man bara vet namnet på ett program som man vill ha hjälp om kan man skriva `man program` och får då en utskrift av en hjälpsida i terminalen. För att få nästa sida trycker man `[space]` och för att komma tillbaka till förra `[b]`, avslutar med `[q]`. Det finns fler sätt att använda `man` men nu vet du ju vad det heter så det är bara att skriva `man man;-)`. Den vanligaste varianten är `man -k söksträng` som letar igenom den korta beskrivningen i alla mansidor efter strängen `söksträng`.

Numer håller GNU på att gå över till hjälpkommandot `info` som är lite modernare och har en hieraktisk struktur där man kan hoppa mellan länkar till andra närgränsande sidor med hyperlänkar. Man går med markören med piltangenterna till en av raderna som börjar med en stjärna med piltangenterna och trycker `[enter]` för att följa länken. Avslut med `[q]` precis som för `man`. `info info` ger förstås mer information. Ubuntu-systemen innehåller en blandning mellan de två systemen.

4.2 Filsystem och användare

POSIX-system är från grunden gjorda för fleranvändarsystem och har mycket djupt rotade användarbehörigheter för filer. Det gör att t. ex. virus är ett helt okänt begrepp. Det intuitiva att tänka sig en fil i en dator är en mängd data som är lagrat på en disk eller liknande lagringsenhet. I POSIX filsystem är det mycket mer än så som behandlas som filer. Den inkommande strömmen signaler från tangentbordet är till exempel en fil. Den går förstås bara att läsa från men inte att skriva till, det omvända gäller för den ström som skicka till skärmen eller en skrivare, men även de är filer i operativsystemet. Varje fil har en ägare som är en användare på datorn och tillhör en grupp som är en samling av användare. Varje fil har sedan separata regler för ägaren, gruppen och övriga. För var och en av dessa kategorier är det angivet om de får läsa, ändra och köra filen. En katalog i filsystemet är bara en fil som innehåller en drös länkar till andra filer, samma regler gäller för dem, om man har behörighet att skriva till en katalog betyder det alltså att man kan lägga till nya filer till den, och för att överhuvudtaget få se vilka filer som ligger i den krävs att man har rättigheter att läsa den. Se t. ex. http://en.wikipedia.org/wiki/File_system_permissions för en genomgång av hur det fungerar.

Det går att montera filsystem från t. ex. ett windowssystem till ett linuxsystem men i de filsystem som används i windows finns det inte så detaljerade kontroll på behörighet och man måste då välja hur man ska översätta dem. Antingen måste man ge ganska fria rättigheter och får då med en del av windows alla säkerhetsproblem eller så får man vara restriktiv och då får man i stället ett filsystem som man nästan inte kan göra något med.

Eftersom det alltid är linuxkärnan i som måste öppna filer och rättighetssystemet ligger i kärnan är det väldigt svårt att göra något för en inkräktare i ett system för att komma runt detta, utan kärnan stannar datorn och det går alltså inte att byta kärnan medans datorn går.

I ett klassiskt unix-system finns det en allsmäktig användare och det är root. Det normala är att root har ett eget lösenord och att man behöver kunna det för att kunna ändra och läsa i alla filer. Till ubuntu-systemen används ett system som kallas sudo som betyder att en användare kan få rättigheter att göra allt som root får genom att ange sitt eget lösenord, man kan säga att man blir root för att göra just det kommandot och sedan går tillbaka till att ha sina normala rättigheter igen när det är klart. För grafiska program brukar det fungera så att man klickar på en programikon och när det startar får man en ruta att fylla i sitt lösenord. Om man ger kommandon i skalet skriver man helt enkelt sudo före kommandot.

Filnamn som börjar med en punkt brukar kallas dolda filer. De är inte mer hemliga än några andra utan styrs helt av sina rättigheter. Det är bara en konvention som man använder för att kunna sälla bort lite filer från när man visar filer i en filhanterare eller med något textkommando. Det vanliga textkommandot `ls` visar till exempel inte dem om man inte anger växeln `a` d. v. s. `ls -a`.

4.3 Skal (kommandotolk)

Till skillnad från Windowssystem som gått från att vara ett grafiskt gränssnitt på ett textbaserat Operativsystem (OS) till att bli ett helt grafiskt system har POSIX-baserade system en helt textbaserad grund. (Märk att MacOS har gått den motsatta vägen från att ha varit ett grafiskt OS till POSIX.) Det betyder att i princip allt går att göra med olika textbaserade program i textterminaler. En central del i ett sådant system är kommandotolken eller skalet. Det motsvarar den gamla Disc Operating System (DOS) i MS-system, men där tar också alla likheter slut. Skal finns i en uppsjö men det klart vanligaste är det ni får upp som förval i systemen, nämligen Bourne Again Shell (BASH). Det är både en kommandotolk som kör de program man skriver namnen på vid prompten men också ett helt programmeringsspråk. Så fort man behöver skriva kommandon som

blir långa och besvärliga, eller om man behöver upprepa en sekvens av kommandon efter varandra kan man lika gärna ta dem och lägga dem i en textfil och skriva `bash filnamn` vid kommandoprompten och resultatet blir exakt detsamma som om man hade skrivit kommandona direkt och man har vips gjort sitt första BASH-program.

I själva verket är många av de program ni har kört redan skrivna i BASH och interpreteras direkt av skalet. Även för program som uppenbarligen inte är skriva i ett skriptspråk är den fil som man åberopar från skalet ett skript skrivet i BASH som sätter upp parametrar och sen i sin tur startar själva programmet.

Vi ska inte göra någon djupdykning i skal i den här kursen men när man vänjer sig är många saker betydligt snabbare att göra i skal än t. ex. den grafisk filhanteraren, speciellt när man ska göra samma sak i många filer eller ta kontroll över processer. POSIX och BASH är också mycket stabila medans X ibland kan hänga sig eller i alla fall frysa skärmbilden om något program kraschar då räcker det oftast att byta till en textterminal och döda den processen igen.

4.4 Pipelines

Se vidare Walleij 90–96.

Unix och därmed GNU har byggts efter en grundläggande princip som är väldigt annorlunda mot Microsofts operativsystem, nämligen snarast som ett lego eller mekanosystem men mängder av små fristående program som gör en specialiserad uppgift och samverkar till att göra en uppgift.

Om vi tar ett så enkelt exempel som att vi vill ha hjälp med ett kommando så kan man använda kommandot `man`. Prova t. ex. att skriva `man bash` vid kommandoprompten. Då får du upp en hjälpsida för skalet `bash`. Själva den filen där den här hjälpinformationen ligger lagrad finns i en fil `/usr/share/man/man1/bash.1.gz`. För det första ser vi direkt på filändelsen att den är komprimerad med programmet `gzip`. För att titta på innehållet i den filen kan man använda kommandot `zless` som skriver ut innehållet i en komprimerad fil på skärmen. Skriv in `zless /usr/share/man/man1/bash.1.gz`. Som ni ser är det här inte särskilt likt den hjälpsida ni fick upp på skärmen. Den är knappast skriven av en människa i den där formen heller. Det här speciella formatet för mansidor kompletteras med information från vårt system av programmet `zsoelim` och skickas vidare till programmet `groff` som typsätter det. För att anpassa den typsatta filen att visas i ett textfönster används ett program som heter `grotty`. Nu är ju den här sidan ganska lång men det som kommer upp först är så mycket som får plats på skärmen, och om man trycker på olika tangenter som `Return`, `space` eller `b` kan man läsa vidare eller gå tillbaka, det sköts om av ett program som heter `less`. Med lite optioner till programmen får man fram att man skulle kunna få ungefär samma beteende som `man bash` genom att skriva `zcat /usr/share/man/man1/bash.1.gz | zsoelim | groff -mandoc -Tascii -Z | grotty | less`. Prova får du se.

Att sätta ihop en rad program som var och en gör ett litet steg i en kedja på det här sättet kallas att göra en pipeline och tecknet `|` kallas pipe och talar just om att man ska ta det som kommer ut från programmet före och skicka vidare till programmet efter. Var och ett av programmen i den vår långa pipeline ovan kan användas i andra sammanhang och delar kan bytas ut. Det normala för `groff` annars är att skicka sin typsatta text direkt till skrivaren, men eftersom vi inte har installerat skrivarna ännu kan vi se hur det skulle ha sett ut genom att spara resultatet till en fil istället. `zcat /usr/share/man/man1/bash.1.gz | zsoelim | groff -mandoc > test.ps` ger en postscriptfil som heter `test.ps` och som egentligen är ämnad att skickas direkt till en skrivare. Vi kan titta på den med något av programmen `ghostview` eller `gv`. (Tecknet `>` ovan säger att man ska skriva det som kommer ut från programmet före tecknet till filen som står efter.)

Som synes är det hela ganska mycket som att bygga lego och eftersom alla klossar finns till hands är alltså det jobb som man behövde göra bara att ta reda på var filen

finns och sätta ihop klossarna. (Som ni kanske har gissat är en del av de här klossarna i själva verket själva pipelines av andra mindre program.)

4.5 Pakethanterare

Se vidare Walleij sid 150–154.

För att kunna hantera att installera den här mängden av små program till operativsystemet finns till alla distributioner av POSIX-baserade OS någon pakethanterare. De är olika för olika distributioner. Debian och Ubuntu använder systemet Advanced Package Tool (APT) som utvecklats av Debian. Det finns en mängd olika program för att administrera men alla anropar samma grund. `dpkg` används för att installera ett paket som du har som en fil på datorn. `apt-get` används för att installera program från en standardiserad källa som kan vara CD-skivor, DVD-skivor en spegel på nätet, eller ett lagringsmedia på datorn. Under installationen använde installationsprogrammet `apt-get` hela tiden, genom att ta paket från installations-CDn. Om man vet namnet på ett paket man vill installera är det enklaste att ge kommandot `apt-get install paketnamn`. Om man inte vet är det enklare att använda någon frontend, till ubuntu är textbaserade `dselect` eller `aptitude` eller X-baserade `synaptic` vanligast.

4.6 Textredigering

Så fort man vill göra något bakom den glättiga ytan av ordbehandlare och webbläsare i en dator kommer man att behöva ändra i rena textfiler, märk väl att det är stor skillnad på en textfil som man redigerar i en valfri texteditor och ett ordbehandlingsdokument från t. ex. MS Word som i stort inte kan tolkas av något annat program. Det finns två stora klassiska texteditorer i POSIX:

VI (VIvisual editor) som är officiell del av POSIX-standarden är de verkliga nerdarnas favorit, fullständigt obegriplig för dödliga. Det finns bara en sak man behöver lära sig (man kan tyvärr råka ut för system där det är den förvalda editorn och man plötsligt sitter med någon viktig systemfil öppnad i VI) och det är det avsluta: `Esc Esc : q !`. Om man börjar skriva på tangentbordet är stor risk att man har gjort massor med ändringar och sparar dem till filen innan man vet ordet av.

EMACS (Editor MACroS) (eller enligt onda tungor Esc Meta Alt Ctrl Shift om du läser Walleij förstär du varför) är inofficiell men finns nästan alltid, men konstigt nog inte i första basinstallationen av Debian så vi får installera själva. EMACS är lite av sin egen värld med ett speciellt programmeringsspråk EMACS LISP för att konfigurera den till att göra allt möjligt (filhanterare, webbläsare, epostklient...). Tyvärr därför lite svårt att göra enklare inställningar. Se vidare Walleij sid 84–89.

Enklare och mer intuitiva texteditorer som kommer med installationen är X-baserade Mousepad och textbaserade pico.

4.7 Andra användarprogram

De program som vanliga brukare normalt kommer mest i kontakt med är väl kanske webbläsare, ordbehandlare, kalkylprogram och ritprogram. Alla de här programmen använder mycket grafiska saker i gränssnitten, (*widgets*) det är knappar, menyer skrollister osv. De två stora skrivbordsmiljöerna Gnome och Kool Desktop Environment (KDE) har varsin helt egen sådan uppsättning Gnomes kallas GTK+ och KDEs kallas Qt. Båda miljöerna har sviter med program för sådant här typiskt kontorsbruk var och en baserad på sin uppsättning. För datorer med lite primärminne är det en viss fördel att använda program

från den skrivbordsmiljö man använder för att inte belasta minnet med att ladda in dubbel uppsättning av alla grafiska element. I tabell 1 är några vanliga tillämpningsprogram listade.

	Gnome	KDE	andra
Texteditor	Gedit	Kwrite	mousepad
Webbläsare		Konqueror	Mozilla, Firefox, Opera
Filhanterare	nautilus	Konqueror	Thunar
PDF-läsare	evince		Acrobat, gv, xpdf
Kontorssvit		Koffice	Openoffice
Ordbehandlare	Abiword	Kword	OO Writer
Kalkylprogram	Gnumeric	Kspread	OO Calc
Presentation		Kpresenter	OO Impress
Bildbehandlig	GIMP	Krita	
Ritprogram	Dia, Inkscape	Karbon14	OO Draw

Tabell 1: Några vanliga program. Openoffice är en populärare kontorssvit för Gnomemiljön än GTK+-programmen och GIMP är den populäraste bildbehandlaren i KDE.

5 TCP/IP, Nätverk

Mål Efter dagens övningar är det meningen att ni ska

1. förstå den principiella uppdelningen i olika nivåer av kommunikationsprotokoll i TCP/IP-stacken och
2. vad som menas med
 - (a) paket,
 - (b) portar,
 - (c) server,
 - (d) nätverksprotokoll,
 - (e) demoner och
3. kunna starta upp en server.

Läs

- <http://www.lysator.liu.se/~kjell-e/tekla/linux/security/tcpip.html>
- Nedanstående

Övningar Ö31 Gör en uppkoppling till en annan dator.

1. Kontrollera att ni har Secure SHell (SSH) installerat med kommandot `which ssh`. Om det inte är det, installera paketet som heter `ssh`. (`sudo apt-get install ssh`)
2. Kontrollera att `ssh`-demonen som svarar på anrop från andra datorer kör. `ps aux | grep sshd` Om inte, starta den `sshd`.
3. Slå er ihop med en annan grupp. Skapa ett konto åt dem på ert dator med `sudo user-add` och se till att ni får ett på deras.
4. Kontrollera vilket IP-nummer ni blivit tilldelade med `ifconfig` och meddela den andra gruppen.
5. Använd `ssh` för att logga in och få ett skal på den datorn.
6. Se om ni kan öppna grafiska program genom att försöka med t. ex. `xeyes`. Om ni inte kan det, läs manualsidan för `ssh` och se vad ni ska göra för att få tillgång till den funktionen och prova igen.

Ö32 Använd kommandot `scp` för att koppla skicka över filer till ert konto på den andra datorn.

*Ö33 Sätt upp SSH-kommunikationen mellan era datorer på en annan port än den standardiserade.

*Ö34 Se om ni kan få upp en helt grafisk miljö från den andra datorn med XDMCP. Det kan kräva att man gräver lite i konfigurationsfiler framför allt på den dator som ni ska logga in till. Använd hjälpen för XDMCP och sök på nätet.

Ö35 Kontrollera vad som egentligen kommer ut på ett vanligt anrop till en webbserver. `Telnet` är ett enkelt protokoll som tidigare användes till kommunikation mellan datorer. Nu kan man använda det för att ansluta till olika portar på en dator och se vilka kommandon den svarar på och vad som kommer ut. Prova att ansluta till serverns `http`-port (nr 80) med `telnet 10.10.10.1 80` när du fått kontakt ge kommandot `GET`.

Ö36 Kontrollera vilka portar ni har öppna med `netstat`. Sök reda på information om dessa. Är det rimligt att de ska vara öppna?

Ö37 Skanna av vilka portar era kompisar har öppna med `nmap`. *Gå inte utanför vårt lokala nät 10.10.10.xxx när ni skannar datorer.*

- Ö38 Ta reda på vilka routrar som används på vägen till olika datorer med `tracert`. Ta hjälp av manualsidan för att tolka resultaten. Är det några adresser där man inte får fram alla stegen? * Går det att få bättre resultat till dessa med andra portar och protokoll?
- Besvara** Q9 <http://www.ipnummer.se/> är en tjänst som visar vilket IP-nummer du har, vad visar den? Stämmer det överrens med ni får från `ifconfig`? Varför blir det så?
- *Q10 I distansuppgiften anges adresser för SSH och Hyper Text Transfer Protocol (HTTP) som inte är standardportar. Det är vår server borta i skrubben som uppgifterna ska ligga på och det är inte för att göra det besvärligare för er jag kommit på att byta portnumren utan för att jag var tvungen, varför? (Det har en koppling till förra frågan.)
- *Q11 Om man sätter upp ett eget Local Area Network (LAN) som är anslutet till internet via en router, spelar det då någon roll vilka IP-nummer man använder. Vilka ska man i så fall använda?
- *Q12 Vad menas med Denial of Service (DoS)-attacker? Hur fungerar de och vilka datorer står oftast för själva attacken.

5.1 TCP/IP-stacken

För att ordna trafiken i datornätverk måste den struktureras upp mycket. Det görs i några olika nivåer som jag ska försöka förklara här. Varje nivå har några olika protokoll, dvs överrenskomna standarder som de använder för att kommunicera. Det sätt att strukturera upp det här idag är nästan uteslutande den sk TCP/IP-stacken som använder fyra nivåer.

5.1.1 Länk

Först och främst behövs den lägsta nivån för hur all trafik på som går genom en fysisk anslutning ska vara strukturerad, det kallas länkskiktet. Våra datorer är anslutna via Ethernet och era bärbara använder Wi-Fi här i skolan. För uppringda modem används PPP och det finns några till. Det här lagret ser bara till att två noder (datorer eller nätverksutrustning) kan prata med varandra. Vid en kommunikation mellan två datorer som inte sitter ihop direkt med varandra kan det därför användas flera olika länkprotokoll längs vägen.

5.1.2 Nätverk (IP)

Det finns några olika protokoll i nätverksskiktet men det som totalt dominerar är Internet Protocol (IP).

Vid användning av IP överförs små paket med information och med uppgift om avsändar- och mottagaradress i form av IP-nummer. (Ett exempel på ett IP-nummer är 213.65.243.16.) Informationen kan överföras via flera alternativa vägar inom nätverket, varje enskilt paket söker sin väg. IP garanterar inte att informationen kommer fram till mottagaren, utan ansvaret för detta ligger på det aktuella protokollet i transportskiktet. Ett lokalt nätverk avgränsas av en nätverksmask som anger om transporten är lokal eller ej. Om mottagaren ligger utanför det lokala adressesegmentet krävs att transporten sker med en Level-3 router.

Den aktuella versionen av IP heter IPv4 (Internet Protocol Version 4). Nästa version, som började att utvecklas 1991 av Internet Engineering Task Force (IETF), heter IPv6 (Internet Protocol Version 6). Specifikationen var klar 1997 och lanserades i juli 2004, då ICANN lade in IPv6-poster i sina DNS-datorer för .jp (Japan) och .kr (Korea).

IP version 4 (IPv4) är den nuvarande versionen av IP. I IPv4 består ett IP-nummer av 32 bitar, vilket är ekvivalent med 4 bytes och gör att antalet adresser är begränsade

till drygt 2 miljarder. Vanligen anges IP-numret genom att skriva ut de 4 byten i decimal form med punkter emellan där varje siffergrupp motsvarar en byte. Många program kan hantera båda sätten att skriva inklusive ett tredje oktalt skrivsätt. Istället för tal anges Internetadresser ofta som en textsträng (domännamn), vilka sedan vid behov automatiskt slås upp i en stor katalog (en Domain Name Server (DNS)) till ett IP-nummer.

För att adresserna i IPv4 ska räcka krävs det att vissa tilldelas dynamiskt. Adresserna kan på det sättet användas av flera kunder, dock inte samtidigt. De s.k. fasta IP-numren tilldelas åt de datorer eller lokala nätverk som har fast uppkoppling som bredband eller egna anslutningar till företag o.s.v. De dynamiska IP-numren tilldelas oftast hemanvändare som får ett nytt nummer varje gång de kopplar upp sig mot Internet.

Vare sig statisk eller dynamisk så tilldelas adressen när datorn slås på (via Dynamic Host Configuration Protocol (DHCP)) eller när datorn ringer upp en modempool (via Point to Point Protocol (PPP).) IP-numret delas ut av internetoperatören såvida den inte används i ett lokalt nätverk. Det finns tre fördefinierade block som kan användas för lokala nätverk.

Felhantering IP-paket kan irra bort sig i nätverket om routrarna på vägen har dålig koll på var de ska skicka dem vidare eller om inte mottagaren är tillgänglig. Datorn kan ju vara avslagen eller ha nätverksproblem. För att inte paketen ska irra runt i evighet mellan olika routrar och fylla upp internet med skräp finns det en nedräkningsprocess inbyggd. När man skickar ett paket anger man hur många routrar det maximalt får använda innan det är framme. Varje router som tar hand om paketet och skickar det vidare minskar den här siffran med ett. När det kommer ner till noll kastas paketet bort.

För att hålla reda på datapaketerna i nätverket finns även ett särskilt protokoll för övervakning, Internet Control Message Protocol (ICMP). När t. ex. ett datapaket kastas bort för att det tagit för lång väg till sitt mål skickar den router som gjorde det ett paket tillbaka till avsändaren med information om att paketet försvann. ICMP-paketerna har inte lika hög prioritet som datapaketerna och det skickas t. ex. ingen information om när de kastas bort av samma skäl.

Normalt finns ingen information om vilken väg paketen går mellan två punkter men om man vill reda ut det kan man använda tricket att skicka ett paket som man säger inte får passera någon router alls på vägen. Den första routern man kommer till kommer då att kasta bort det och skicka tillbaka ett felmeddelande. Så kan man skicka ett till paket mot samma mål och säga att det får passera en router på vägen och på så sätt få ett felmeddelande från nästa router på vägen. Om man tar tid på hur lång tid det tar innan felmeddelandet kommer tillbaka ser man hur lång tid de olika stegen tar. På så sätt kan man kartlägga vilken väg, eller få statistik på vilka vägar, om de ändras som paketen tar. Det kan vara intressant att se om man t. ex. tycker att man har dålig anslutning var flaskhalsen sitter. Ett program till linux som fungerar efter den här principen är traceroute, till windows finns tracert som gör samma sak.

5.1.3 Transport

Transportskiktet håller reda på hur de små paketen som skickas ska behandlas. Det talar bland annat om vilken port som ska paketet ska till på den mottagande datorn.

Det klart vanligaste som används för att föra över filer (t. ex. när man surfar på nätet som ju handlar om att ta hem en massa filer till datorn) är Transmission Control Protocol (TCP). Det ser till att de paket som kommer fram till mottagaren inte har ändrats på vägen genom en kontrollmekanism. Det ser också till att alla paketen har kommit fram och sorterar upp dem i ordning. En fil är oftast uppdelad på en mängd paket och eftersom de kanske har tagit olika vägar är det inte säkert att de kommer fram i samma ordning som de skickades. När de sorterats kan de sättas ihop till en fil igen. Om något paket i serien saknas skickar den mottagande datorn en fråga till den sändande att skicka det igen tills det finns en komplett samling.

För en del tidskritiska överföringar har man inte tid att samla ihop och sortera upp paketen, då används det mycket enklare User Datagram Protocol (UDP) som inte har dessa funktioner utan bara kastas ut på nätverket på vinst och förlust. Eftersom det används just för tidskritiska överföringar brukar de ofta ges högre prioritet i nätverket. Exempel på användningar är tidsangivelser från en tidserver för att ställa klockan på datorn, det är uppenbarligen värdelös information att fråga om, men även i en del actionspel och röstöverföring med IP-telefoner får man hellre tappa bort en del paket och än att det ska bli uppehåll och hacka.

5.1.4 Tillämpningar

Den fjärde och högsta nivån är tillämpningslagret eller applikationslagret. Det är det här man normalt kommer i kontakt med som användare. Det är det protokoll som programmen man använder har för att prata med varandra. Vanliga protokoll är HTTP som används för World Wide Web (WWW), Internet Relay Chat (IRC) som används för chatt, File Transfer Protocol (FTP) för filöverföringar, Bittorrent för fildelning och SSH för krypterade överföringar. Även bakom kulisserna används en del tillämpningsprotokoll för att få nätverket att fungera t.ex. används DNS för att fråga efter vilket IP-nummer som motsvarar ett datornamn.

5.2 Datornamn

Varje enskild dator i ett nät har ett IP-nummer, det är ett unikt nummer i det nätet så att nätverksutrustningen kan skicka paketen till rätt dator. Eftersom det är svårt för människor att komma ihåg många långa nummer kan man i stället för IP-numren ofta ange ett datornamn. Det är då avgränsat med punkter så att den sista delen ger den toppdomän som datorn tillhör, på internet är de t.ex. se för sverige com för kommersiella företag och org för organisationer, till vänster kommer mer detaljerad information om underdomäner och före den första punkten kommer själva namnet på datorn. För att få tag på den dator som har namnet måste någon hålla reda på vilket namn som motsvarar vilket nummer, denne någon kallas Domain Name Server (DNS). En fördel med att använda namn istället för IP-nummer är att man enkelt kan byta namn på datorer i ett nätverk och att en dator kan ha flera olika alias (smeknamn) i DNSen. Om ni tänker på datornamn ni brukar se så är det väldigt ofta datorer heter www. I praktiken är det förmodligen väldigt få datorer som har det namnet, det är ett typiskt alias som man använder på den dator som man använder som webserver. Om man sen vill byta webserver så kopplar man in en ny dator på nätverket sätter upp en webserverdemon på den och när allt fungerar som man vill byter man bara ut aliaset www i DNSen så att det pekar på den nya datorn. På stora webbplatser som har mycket trafik, (sökmotorer, stora tidningar etc.) kan inte en dator ta hand om alla förfrågningar, då låter man DNSen ge olika IP-nummer varje gång till en massa datorer som allihop kör varsin webserver med samma innehåll som får dela på bördan.

5.3 URL

En Uniform Resource Locator (URL) är adressen till en fil eller tjänst på nätet. Den består normalt av först en del som talar om vilket applikationsprotokoll som används följt av ett kolon och två snedstreck (`http://`, `ftp://` etc.). Det som kommer därefter beror på protokollet men för HTTP kommer namnet eller IP-nummret på datorn som har tjänsten eller filen avgränsad av punkter (`www.fhsk.kiruna.se`, `192.168.87.90` etc.) och därefter en sökväg på datorn avgränsad av snedstreck (`/~johan/`, `/pub/os/linux` etc.). Det är alltid de här avgränsarna som gäller, idag har det blivit ganska vanligt med sk. nätfiske (phishing). Lurendrejare vill av någon anledning försöka ge sken av att vara någon de inte är på nätet t.ex. en känd bank. Då gäller det att ha en adress som ser trovärdig ut och det

de ofta luras med är att någon av avgränsarna är utbytt, <http://www.bank.se/sv/login> är en dator i toppdomänen El Salvador medans <http://www.bank.se/sv/login> är en i toppdomänen Sverige!

5.4 Portar

Begreppet portar på datorer är lite tvetydigt. Från början när en dator var en isolerad burk som man ville ansluta något till gjorde man en kontakt där det gick att anluta detta något. Det gick då att sätta en sak där som skickade sina signaler via kontakten och den kallades en port. Exempel så sådana är serieporten och parallellporten som t. ex. användes till skrivare.

När man började sätta ihop datorer i nätverk uppstod problemet att man ville använda samma kabel mellan datorer för flera olika program på de olika datorerna och helst dessutom samtidigt eller i alla fall blandade efter varandra. I transportprotokollen i IP-stacken finns det därför med i paketen en siffra som är en signal om vilket program på den mottagande datorn det är till. Tillämpningsprotokollen har särskilda standardvärden för dessa så man behöver sällan bekymra sig om dem. Det finns dock en del tillfällen då man vill frånga standarden, speciellt om man kör flera program av samma sort måste man kunna skilja på vilket det är till, då säger man åt det ena av dem att lyssna på ett annat portnummer än det normala.

Eftersom olika tillämpningsprogram har olika stor säkerhet mot attacker och avlyssning används portnumret ofta för att bestämma vilka paket som ska få komma igenom en brandvägg.

5.5 Demoner

För att två datorer ska kunna prata med varandra krävs både att en pratar och att den andra hör när någon ropar på den. Program som sickar ut förfrågningar och alltså 'pratar' är vanliga program ni är vana att använda t. ex. webbläsare, de startar man när man vill använda dem. De program som lyssnar måste däremot göra det hela tiden, de kan ju inte veta när någon ska ropa. Att starta ett sånt program på sin dator kallas att sätta upp en server. (Alltså ett till ord som har flera betydelser, en server kan vara en fysisk dator som står i en datorhall och har till uppgift att svara på sådan anropstrafik men också en tjänst som en dator tillhandahåller.) Programmen som man kör när man har satt upp servern kallas demoner. De ligger i datorns minne och tar plats och ska därför vara så små som möjligt och efter de är startade behöver de inte använda processorn förens det kommer ett anrop på den port som demonen är satt att lyssna på. Det som då händer är oftast att demonen delar på sig i två processer, en svarar på det anropet som kom och håller kontakten med den som ropat, de sätter upp en egen länk mellan sig, och den andra går tillbaks till det föregående läget och fortsätter att ligga och lyssna på nya anrop.

6 Webbserver, HTML

Vi börjar med att skriva dokument i **HTML** och installerar så webbservern **Apache**.

Mål Målet med dagens övningar är att alla ska kunna skriva en webbsida, och lägga ut den på nätet med en webbserver.

Läs Nedanstående

- Övningar**
- Ö39 Gå igenom <http://www.w3.org/MarkUp/Guide/> och gör en HTML-sida som har en överskrift, innehåller en lista och/eller tabell, en länk till en annan webbsida och en bild. Sidan ska vara strukturerad med doctype head och body. Använd din favorittexteditor för att skriva koden och kontrollera hur den ser ut genom att öppna filen med webbläsaren.
 - Ö40 Gör en HTML-sida till och gör en länk till den, prova att det fungerar.
 - *Ö41 Använd W3Cs valideringstjänst <http://validator.w3.org/> för att se att sidan är korrekt skriven.
 - *Ö42 Se om du har någon What You See Is What You Get (WYSIWYG) HTML-editor, (mozilla composer, Quanta), prova att börja om från början med att skriva en likadan sida som du gjorde i övning Ö39. Titta på resultatet i en webbläsare, jämför HTML-koden genom att öppna filen i texteditorn, använd 'visa dokumentets källa', 'visa källkod' eller liknande i webbläsaren eller använd kodediteringsläget i HTML-editorn om det finns ett sådant. Vad finns det för skillnader mot den ni skrev själva?
 - *Ö43 Använd Cascading Style Sheets (CSS) för att ändra utseendet på sidan. HTML.net har en tutorial på <http://www.html.net/tutorials/css/>, action börjar på lesson 2. En annan är <http://friendlybit.com/css/beginners-guide-to-css-and-standards/>. Ändra färgen på överskriften på sidan till navy, bakgrunden till någon annan ljus färg än vit. Byt typsnittet på texten till verdana. Se själv vad du kan göra mer.
 - Ö44 Gör en katalog som heter `public_html` direkt i er hemkatalog (d. v. s. med det totala namnet `\home\användarnamn\public_html`).
 - Ö45 Flytta filen ni gjorde i övning Ö39 till katalogen `public_html` som ni just gjorde och döp den till `index.html`.
 - Ö46 Skriv in adressen <http://localhost/~användarnamn> i webbläsaren och se om ni får upp något.
 - Ö47 Installera webbservern. Välj att installera paketet `apache2` och `php5`, då bör ni få en lista beroenden. Installera den listan ni får. (Om ni måste välja mellan `libapache2-mod-php5` och `php5-cgi` ska ni ta det första.)
 - Ö48 Jag är osäker på om distributionen har en konfiguration där modulen för att visa användarnas kataloger laddas. Skriv in adressen <http://localhost/~användarnamn> i webbläsaren igen och se vad som händer nu, om ni ser er sida är det klart annars måste man själv lägga till länkar till modulerna. Det görs genom att gå till katalogen `/etc/apache2/mods-enabled` där moduler som ska laddas ligger med kommandot `sudo cd /etc/apache2/mods-enabled` som rootanvändare. Där ska man göra en länk till modulen som ligger i `/etc/apache2/mods-available/`, det är två filer som ska länkas, `userdir.conf` och `userdir.load`. Länkarna görs med kommandona `sudo ln -s ../mods-available/userdir.conf .` och `sudo ln -s ../mods-available/userdir.load .`

- Ö49 Skriv in adressen `http://localhost/~användarnamn` i webbläsaren igen och se vad som händer nu. Den här gången borde ni om ni följt instruktionerna se sidan ni har skrivit. Tecknet `~` (tilde) i adressen betyder att användarnamn är en katalog i `/home/`. Webbservern är, eventuellt efter vår egen konfigureringskonfiguration, konfigurerad så att den letar efter och visar innehållet i den speciella katalogen `public_html`, (det vore ju inte helt lyckat om den la ut allt som låg i hemkatalogen på webben så att vem som helst får se). Om adressen slutar med namnet på en katalog och inte en fil letar den efter filnamnet `index.html`.
- Ö50 Servern ni använde för distansuppgiften kör en webserver och ni har konton där. Skapa en katalog `public_html` i den och kopiera över era filer. Servern har från folkhögskolans nät IP-nummer 192.168.87.90, och utifrån internet 83.150.146.24, då med HTTP-porten 8080 och SSH-porten 666. Kolla att det går att se era sidor.
- Besvara** *Q13 Vad är skillnaden på HTML och eXensible Hypertext Markup Language (XHTML)
- *Q14 Vilka strategier ska man ha för att skydda sig mot att bli av med data under följande situationer?
1. Strömavbrott
 2. Program eller operativsystemet hänger sig.
 3. Hårddiskkrasch
 4. Att man av misstag skriver över eller raderar filer.
 5. Någon gör intrång i datorn och ändrar på innehållet i filsystemet.
 6. Någon har obemärkt gjort intrång i datorn och kanske ändrat på innehållet under obestämd tid.
 7. Datorn blir stulen eller totalförstörd.
- Finns det någon lösning som täcker in samtliga ovanstående problem?
- Redovisa** • Lägg upp sidorna som ni skapat i övningarna Ö39 och Ö40 på ert konto på vår gemensamma server.

6.1 Webb

WWW är ett system av sammanlänkade hypertextdokument som är åtkomliga för datorer anslutna till internet. Det uppfanns av Tim Berners-Lee vid Conseil Européen pour la Recherche Nucléaire (CERN) 1991. Webben har sedan utvecklats med en del och innefattar en mängd olika filformat. För att standardisera dessa har det bildats ett internationellt standardiseringsorgan World Wide Web Consortium (W3C) som leds av Tim Berners-Lee, alla organisationer kan bli medlemmar i W3C och vara med och bestämma hur standarden ska se ut men medlemsavgiften är ganska hög så det är främst regeringar och stora företag och organisationer som är medlemmar.

6.2 HTML

HTML är ett strukturerat språk för att presentera i första hand text och bilder. Formateringen av innehåller görs med sk taggar, t ex en huvudrubrik görs genom att skriva `<h1>Rubrik</h1>`. Det är `<h1>` och `</h1>` som är taggarna. De kommer oftast i par där sluttaggen är lika som öppningstaggen förutom att de börjar med ett snedstreck.

6.2.1 Hypertext

Med hypertext menas att man kan länka ihop dokument genom att göra vissa ord (eller bilder eller andra element) har funktionen att de innehåller en adress till ett annat dokument eller en annan plats i samma dokument. Det har man dels för att man länka in referenser till helt andra dokument som ligger på internet, dels använder man dem för att strukturera upp sin webbplats. Det typiska användandet är att man har en första presentationssida med en rubrik som visar var man kommit och en innehållsförteckning över vad som finns på webbplatsen i form av länkar.

6.2.2 Struktur vs. design

Det som är utmärkande för HTML är att den som skriver dokumentet kan indikera vad hon vill ha fram för effekt men det är webbläsaren som användaren använder för att visa det som bestämmer hur det presenteras på skärmen. Det skiljer jobbet att forma ett dokument för en webbsida från att göra ett dokument som ska skrivas ut på papper genom att på ett papper kan man bestämma precis hur det ska se ut medans en webbsida får man se till att den är väl strukturerad och lita på att webbläsaren visar det på ett bra sätt för läsaren. Ett vanligt misstag när man skriver webbsidor är att man försöker detaljstyra hur presentationen ska se ut med mängder av HTML-taggar och tittar på resultatet i en webbläsare och tycker att det blir snyggt och inte tänker på att resultatet kan bli helt annorlunda i en annan webbläsare. För att råda bot mot detta har man tagit fram ett strukturerat sätt att skriva önskemål om hur den grafiska presentationen ska se ut i det komplementerande språket CSS. Att skriva sådana lämnas till den intresserade läsaren.

6.3 Webbserver

En normal webbsida i ren HTML går att titta på direkt genom att öppna filen i webbläsaren. För att någon på en annan dator ska kunna se sidan som ligger på din dator behövs en webbserver. Den enklaste webbservern är en demon som lyssnar efter anrop på port 80 och bara svarar med att skicka ut filer helt opåverkade.

Vi ska använda den absolut vanligaste webbservern – Apache. Den kan fås att göra mycket mer än att bara skicka ut filer. T. ex. kan den interpretera program som är skrivna i PHP Hypertext Preprocessor (PHP) och skicka ut resultatet i HTML.

För att ge snabb respons startar Apache flera demonprocesser, när ni har fått igång apache kan ni se med `ps aux`. Om man behöver starta om webbservern måste man skicka en term-signal till huvudprocessen, processnumret på den står finns i en fil som heter `/var/run/apache2.pid`. Det gör man med kommandot `kill processnummer` Sen kan man starta den med kommandot `apache2`.

7 Programmering, PHP

Mål Vi ska lära oss att skriva in PHP-funktioner till våra hemsidor.

Läs Introduktion i <http://se2.php.net/manual/sv/>.

- Övningar**
- Ö51 Gå till http://www.w3schools.com/php/php_syntax.asp, läs och stega er fram till 'variables'. Ändra namnet på filen ni gjorde i övning Ö39 till `index.php`. Börja att titta på sidan och se att ni fortfarande ser samma sida. Gör ett kodstycke PHP i sidan där ni lagrar en textsträng i en variabel och sedan skriver ut den med kommandot `echo`.
- Ö52 Stega er vidare och läs i lektionen på w3schools till 'If...Else'. Lägg till ett nytt PHP-kodstycke där ni gör en ny variabel `$foo` med ett numeriskt värde. Gör en if-sats som skriver ut "värdet är mindre än tio" om värdet på variabeln är mindre än 10. Prova med att ändra värdet på variabeln i koden om de två fallen fungerar. Komplettera med en else-sats som skriver ut "värdet är större än 10". Nu blir det ju tokigt om ni ändrar till just 10 så skriv slutligen in en elseif-sats för specialfallet när variabeln har värdet 10.
- Ö53 Ändra namnet på variabeln i if- och elseif-satserna från `$foo` till `$_GET['foo']`. När ni nu ska titta på filen i webbläsaren skriver ni till `?foo=10` efter filnamnet. Prova att ändra värdet i webbläsaren.
- Ö54 Gör en googlesökning efter 'foo' och titta i adressraden i webbläsaren. Det här behöver inte betyda att google är skrivet i PHP för det här är ett standardiserat sätt att skicka med variabler som används av flera programmeringsspråk och som ingår i HTML-standarderna.
- Ö55 Gå igenom 'Enkel lektion i PHP' på <http://se2.php.net/manual/sv/>. Skriv två sidor där den ena har ett formulär som innehåller två inmatningsrutor och ett som skriver ut det som man matar in i formuläret.

8 Programmering fortsättning

Mål Vi ska lära oss att hantera grunderna i programmering, variabler, funktioner, operatorer och styrstrukturer. Börja på ett programmeringsprojekt som ska lösas på kommande lektioner.

Övningar Ö56 Gör de återstående basövningarna på w3schools, gå till http://www.w3schools.com/php/php_arrays.asp och gå vidare fram till 'Functions'.

Ö57 **Projekt** gör *något* av följande:

1. Börja på en deluppgift i ett samarbetsprojekt med flera i klassen.
2. Börja på ett projekt du själv tycker är lagom stort och skulle vilja göra, presentera strukturen och upplägget för läraren.
3. Börja på ett projekt tilldelat av läraren.

Rep.-frågor Q15 Rangordna de olika minnesenheterna i/till en normal dator i storleksordning. USB-minne, hårddisk, L1-cache, L2-cache, RAM.

Q16 Rangordna de samma i läshastighet.

Q17 Vilka parametrar brukar användas för att beskriva en processors prestanda?

Q18 Om man kan välja mellan två i övrigt likvärdiga processorer där en av dem har dubbla kärnor med 32 bitars bussbredd och den andra har enkel kärna och 64 bitars bussbredd, vilka är för och nackdelarna för de två? Vilken skulle du välja?

Q19 Chipset består normalt av en sydbrygga en nordbrygga och en I/O enhet vilken har det snabbaste informationsflödet? Vilken av dem har hand om CD-spelaren?

Q20 Vad heter bussen som går mellan nord och sydbryggan?

Q21 Följande begrepp har alla vars minst två skilda betydelser i datorsammanhang. Förklara skillnaden.

1. port
2. kärna
3. server
4. paket

Q22 Vad menas med demoner? Har ni några demoner i datorn?

Q23 Det finns fyra nivåer i TCP/IP-stacken. Vilken standard använder ni i varje nivå när ni tittar på er egen webbsida från servern här i telefonrummet på er skärm?

Q24 Nämn någon anledning till att man vill dela upp en hårddisk i partitioner.

Q25 Vilken eller vilka av följande säkerhetsmetoder kan hjälpa dig att rädda det arbete du suttit och skrivit hela dagen när hårddisken kraschar på eftermiddagen respektive när du behöver få fram vad du hade på din ständigt föränderliga webbsida förra julen?

1. Spegling till annan dator.
2. Inkrementell backup.
3. RAID
4. versionskontroll

Kommando	Vanliga växlar	Förklaring
ls	-l -a -F	lista filer (<i>list</i>)
cd		byt katalog (<i>change directory</i>)
cp	-r	kopiera fil (<i>copy</i>)
mv		flytta eller döp om fil (<i>move</i>)
rm	-R -f	ta bort fil (<i>remove</i>)
chown		ändra ägare till fil eller katalog (<i>change owner</i>)
man	-k	läs hjälp (<i>manual</i>)
su	-	byt användaridentitet (<i>switch user</i>)
sudo		kör kommando som root-användare (<i>switch user do</i>)
ps	aux -u	lista processer
kill	-9	stäng ner processer
dhclient		begär ny ipadress
scp	-r	kopiera fil till/från annan dator (<i>secure copy</i>)
ssh	-X	logga in till annan dator (<i>secure shell</i>)
more, less		skriv ut texten i en fil så mycket som får plats i fönstret
nano, pico		enkla texteditorer
exit		avsluta skalet (oftast logga ut)

Tabell 2: Vanliga kommandon i POSIX-system.

Examination

Tanken är att examinationen till stor del skall vara att lösa praktiska uppgifter på datorer därför blir det ena halvklassen på förmiddagen och den andra på eftermiddagen. Ni kommer att ha tillgång till den dator ni arbetat med under kursen på provet

9 Programmering avslut, repetition

Mål Få klart våra påbörjade projekt till funktionella webbsidor.

Övning Skriv klart det påbörjade projektet.

Cheklista Se till att följande fungerar på er dator inför slutprovet på fredag:

1. Du har en *egen* användare på datorn och har administrativa rättigheter d. v. s. kan köra kommandon med sudo.
2. Datorn kör en apache webbserver, med stöd för PHP.
3. Datorn har fungerade webbläsare och texteditor installerade.
4. Se till att ni vet vem i gruppen som ska ha prov på förmiddagen resp. eftermiddagen.

Rep.-uppg. Ö58 Se till att du kan komma in i BIOS och kontrollera hur mycket minne BIOS säger att din dator har.

Ö59 Vem har läs- respektive skrivrättigheter till filen /root/.bash_history? Kan det tänkas vara känsligt vad som står i den?

Ö60 Se till att du kan söka reda på och installera ett namngivet paket som finns i ubuntusystemet.

Ö61 Gör en underkatalog till ~/public_html och ändra skrivrättigheterna så att webbservern kan lagra filer där.

Ö62 Se till att du kan skriva textdokument och spara i PDF-format.

Ö63 Ta reda på vilken användare som kör webbservern.

Ö64 Starta om webbservern utan att starta om datorn.

Ö65 Kopiera över en fil från din dator till servern.

Ö66 Gå igenom tabell 2 och se till att du kan använda dem.

A Akronymer

AMD	American Micro Devices
APT	Advanced Package Tool
BIOS	Basic Input/Output System
BASH	Bourne Again SHell
CD	Compact Disc
CERN	Conseil Européen pour la Recherche Nucléaire
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DOS	Disc Operating System
DoS	Denial of Service
DVD	Digital Video Disc
EMACS	Editor MACroS
FTP	File Transfer Protocol
GB	Gigabyte
GNU	Gnu's Not Unix
GPL	General Public License
GRUB	GRand Unified Bootloader
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
KDE	Kool Desktop Environment
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IRC	Internet Relay Chat
LAMP	Linux Apache MySQL PHP/Perl/Python
LAN	Local Area Network
MS	Microsoft
NTP	Network Time Protocol
OS	Operativsystem

PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PDF	Portable Document Format
PGA	Pin Grid Array
PHP	PHP Hypertext Preprocessor
POSIX	Portable Operating System Interface for UNIX
PPP	Point to Point Protocol
PS	Postscript
RAID	Redundant Array of Independent Discs
SQL	Structured Query Language
SSH	Secure Shell
SVN	Subversion
TCP	Transmission Control Protocol
TP	Twisted Pair
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VI	Visual editor
W3C	World Wide Web Consortium
WWW	World Wide Web
WYSIWYG	What You See Is What You Get
XHTML	eXensible Hypertext Markup Language
XML	eXensible Markup Language

Dokumentet typsatt i L^AT_EX.