

Övningar för del I av kursen

Detta är den första samlingen av tre med övningar. Övningarna i den här samlingen numreras med 1.x där x går från 1 till 8. Det är nödvändigt att genomföra de flesta övningar för att kunna få igång arbetet i kursen ordentligt. Vissa övningar ingår desutom som delar i laborationerna som alltså examineras (dvs ger poäng i LADOK). (Det är fallet med 1.7 och 1.8.) Andra övningar leder in i laborationerna. Övningarna kan ni fritt hjälpa varandra med men det är absolut förbjudet att hjälpa varandra med laborationerna – de ska genomföras självständigt.

Övningarna är skraddarsydda för att verkligen komma in i kursen och utgör tillsammans med litteraturstudier, föreläsningssanteckningar och laborationer kärnan i kursen.

Övningar på självstudiematerialet

1.1. Se till att du har *VirtualBox* installerat och körandes den virtuella maskinen *NewTinyDebian*. Maskinen *NewTinyDebian* distribueras i en så kallad OVA-fil som finns på en DVD-skiva som skickas runt på föreläsningen. Man behöver göra *Import Appliance* som finns under *File*-menyn i *VirtualBox*. Sjelva *VirtualBox* hämtas bäst från www.virtualbox.org, då får man den senaste versionen. Det är dock inte alls absolut nödvändigt att ha den senaste versionen.

1.2. Se till att din *NewTinyDebian* har en delad katalog som du väljer på din värdmaskin. Spara alltid alla filer på värdmaskinen, i den delade katalogen, de är tryggare där.

1.3. Prova kommandona som gicks igenom i självstudiematerial 1, `sleep`, `whoami`, `pwd`, `man`, `info`, `ls`, `ls -l`, `cd`, `mkdir`, `rmdir`. Arbeta genom att skapa kataloger med `mkdir` och gå ner i dem, med `cd`, skapa underkataloger där och observera också resultaten parallellt med en filhanterare som ger en grafisk presentation av resultatet. Se speciellt till att du arbetar i den delade katalogen så att du alltså skapar kataloger med `mkdir` på värdmaskinen. Om du använder *Windows* bör du också kunna följa skapandet av kataloger i den delade kataloger med *Windows Explorer*. Om du inte ännu känner till det här med delade kataloger, vänta lite med det till senare.

1.4. Lär dig de nya kommandona `clear`, `echo`, `date` och `mv` genom att läsa manualsidor och undersök hur de kan användas. Exempelvis kan `mv` användas för att flytta och byta namn på kataloger. Testa detta och experimentera tills du känner dig hyfsat trygg med hur alla kommandon fungerar. Undersök också de grafiska presentationerna av resultaten av kommandot `mv` genom att byta namn på en katalog och observera hur namnbytet får effekt, dels genom att använda kommandot `ls`, men också titta i *NewTinyDebian's File Manager* eller *Windows Explorer* eller liknande. Undersök vilka kataloger du rätt att byta namn på med `mv`.

1.5. Knappa igenom några av exemplena från självstudiematerial 2 och läs manualsidor till alla kommandon, `touch`, `rm`, `cp`, `cat`, `emacs`, `gcc` och `python`. Du behöver inte förstå alla detaljer, men skaffa dig ett hum om hur det fungerar. Det är mycket viktigt att du får igång ett C-program alltså att du kan kompilera och köra det.

1.6. Om du inte hittills inte har brytt dig om delade kataloger och bara arbetat under `/home/me`, flytta då allt ditt arbete till en delad katalog istället. Allting ska alltså hamna i den delade katalogen som ligger under `/media/sf_vad_den_nu_heter/`. Flytta filerna hit med kommandot `mv` och observera hur de dyker upp på värdmaskinen vartefter du tar ut dem från den virtuella

maskinen. Alltså, till exempel, om du ska flytta filen `testfil` från `/home/me` till `/media/sf_vad_den_nu_heter/`, observera först att katalogen i värddatorn inte har filen `testfil` på plats i den katalog som delas, utför sedan kommandot `mv testfil /media/sf_vad_den_nu_heter/` ståendes i `/home/me`. Kontrollera *därefter* att filen `testfil` dykt upp i på värddatorn. Utför detta sedan för alla filer som du vill flytta över till värddatorn.

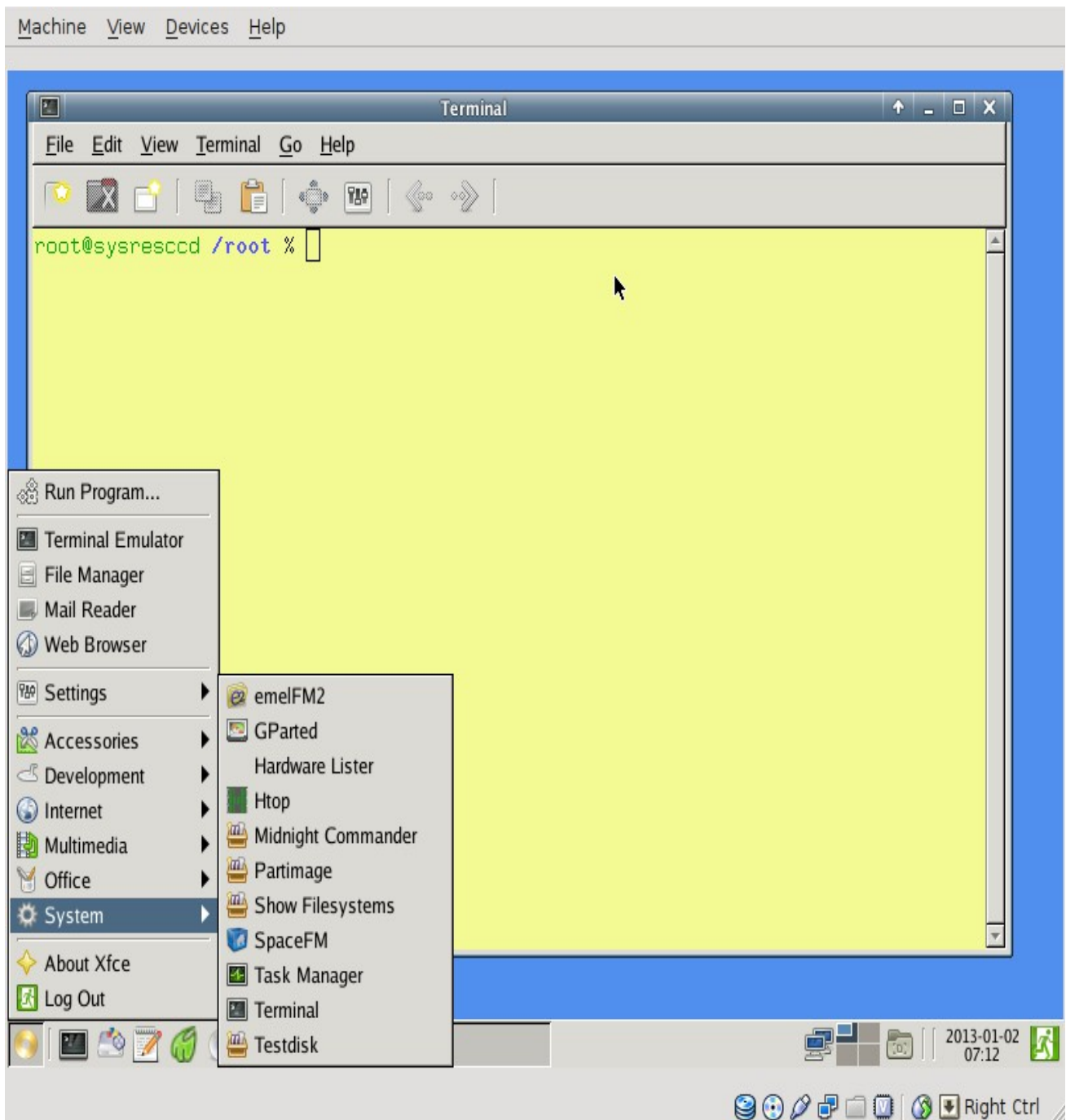
När alla dokument och program du arbetar med på *NewTinyDebian* är överflyttade till värddatorn, ta då för vana att *hela tiden* arbeta utanför den virtuella maskinen. Den virtuella maskinens hårddisk är vald väldigt liten och detta är gjort med medvetenhet, det är inte alltså inte tänkt att ni ska spara några filer i den virtuella maskinen, de ska sparas på värdmaskinen. Det blir ett sätt att separera verktyg och dokument.

Mycket viktig övning: Se till att du har kompilerat och kört C-programmet med hjälp av emacs och gcc

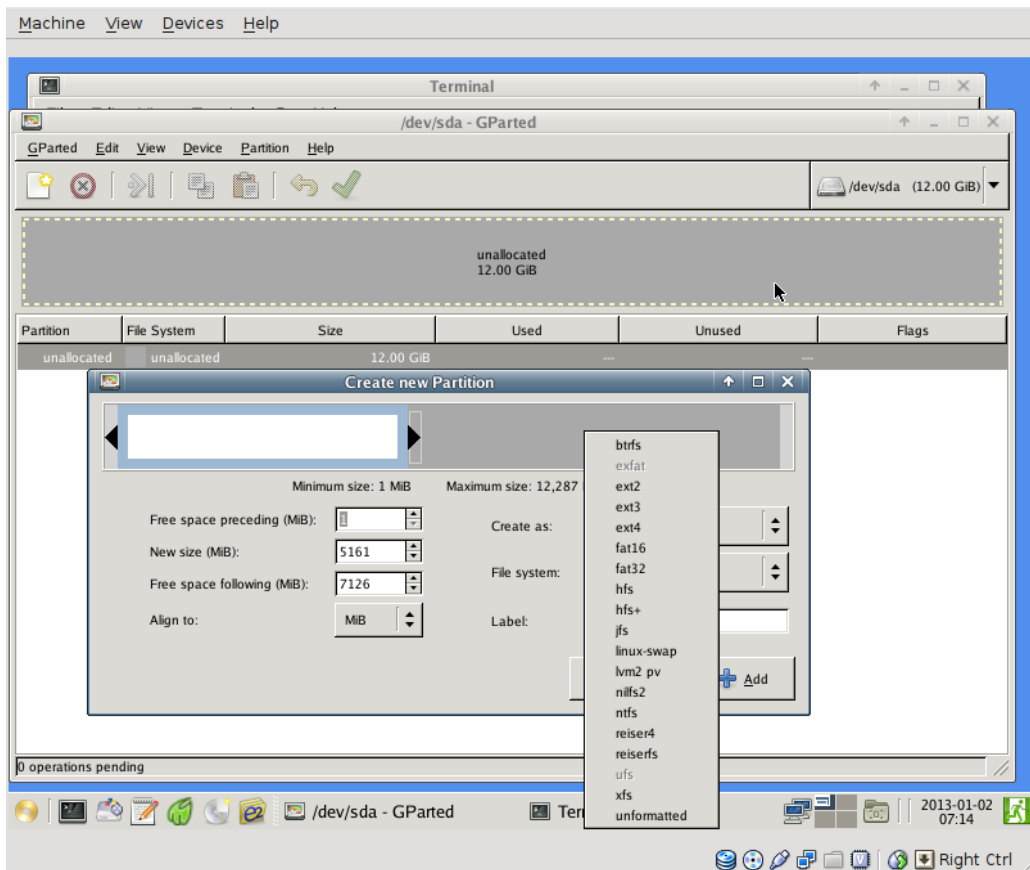
Övningar på kursmöte 1, 2 och 3

1.7. Installation av Debian (detta är en del av laboration 1).

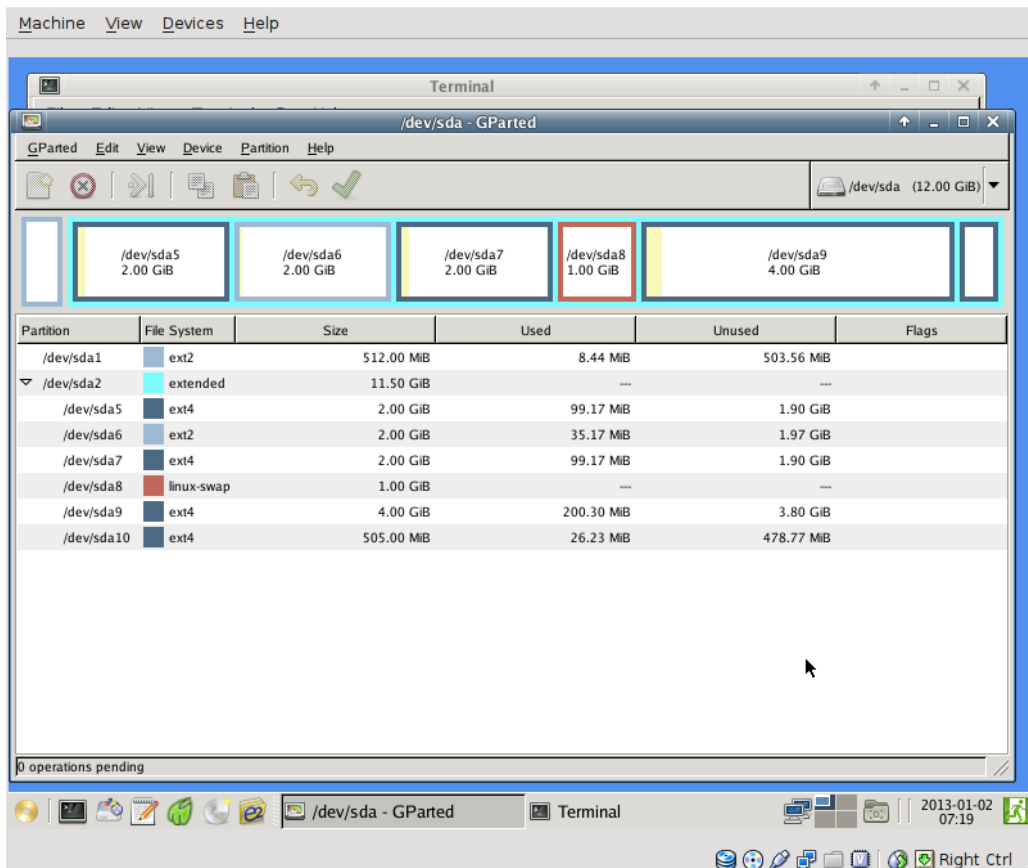
Målet med denna övning är att installera operativsystemet *Debian* på en virtuell maskin så att du har en maskin som liknar *NewTinyDebian* som är en gammal men väl fungerande version. Men vi ska också ta ett lite annat grepp som illustrerar hur man kan använda olika verktyg för att arbeta med operativsystem. Ni behöver hämta någonting som heter *SystemRescueCD* (googla så hittar ni det). Ni behöver sedan skapa en virtuella maskin för installation av *Debian*. Montera sedan *SystemRescueCD* på CD-läsaren och starta systemet. Nu ska ni upprätta ett partitionsschema på den virtuella maskinen. Vi ska då använda verktyget *GPartEd* som ni hittar enligt bilden nedan



Här är precis *SystemRescueCD* startad, vi väljer då programmet *Gparted*, partitionseditorn. Det leder till nästa vy:



Vi kommer in i partitionseditorn, här kan vi, med hjälp av menyer välja hur vi ska partitionera. Efter en del mekande bör det se ut så här:



Egentligen ska vi inte använda detta partitionsschema, eller partitionering, men det är värt göra detta för att få upplevelsen av att ha delat upp en hårddisk i partitioner. Nu kan vi lättare prata om partitioner som kommer att leda in på begreppet *montering* vilket är mycket viktigt inom operativsystemtekniken.

Kommandot för att montera partitioner heter **mount** och man skriver

```
mount -t <typ> <device-nod> <katalog>
```

vid en kommandoprompt där *typ* byts mot den typ som filsystemet har, tex *ext4* eller *reiserfs*, *device-nod* ska bytas mot den partition man vill montera, för oss blir det */dev/sda1* eller */dev/sda5* eller */dev/sda7* eller liknande. Slutligen ska katalog bytas mot den monteringspunkt där man vill att partitionen i fråga (specificerad av *device-nod*) ska monteras in. Arbeta med detta och bygg ihop ett skelett av UNIX-filhierarki i din maskin. Montera roten (som är */dev/sda5*) under */mnt/gentoo*. Och använd I övrigt följande monteringschema:

```
/dev/sda1 på /boot/      /dev/sda5 på /      /dev/sda6 på /tmp/
/dev/sda7 på /var/      /dev/sda8 används som swap (egentligen inte en montering)
/dev/sda9 på /home/
```

När monteringsarna är gjorda ska det gå att gå omkring i strukturen med kommandot *cd*.

Anmärkning: Vi har här beskrivit hur kommandot `mount` används på ett ganska utförligt sätt. Vi kommer fortsättningsvis inte att beskriva de kommandona lika utförligt, det är meningen att ni parallellt med installationsproceduren ska undersöka betydelsen av de manövrar ni utför med kommandot `man` och med Wikipedia. Börja alltså med att göra kommandot `man` på samtliga nedanstående kommandon innan du kör dem.

När du är klar med detta så kan du helt enkelt kasta resultatet, gör lite egna anteckningar och gå sedan på en installation från scratch av operativsystemet Debian. Använd då en installations-CD som du laddar ner från www.debian.org. (Vi har också haft denna image på en DVD på kursmötena, möjligtvis har du den därifrån istället och behöver alltså inte ladda ner.)

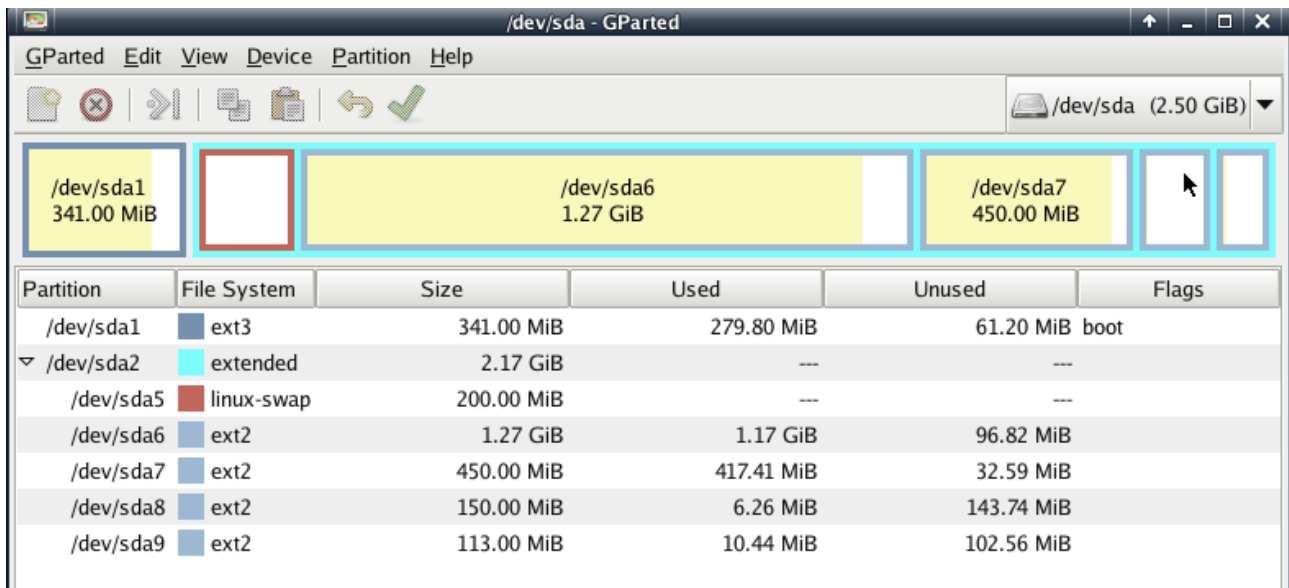
1.8. Omorganisation av *NewTinyDebian*

Hämta maskinen *NewTinyDebian* från kth.box.net (Länk finns på kurswebbsidan, eller så har du redan fått den vis DVD:n). Skapa en kopia av denna maskin, den ena ska vi arbeta med i kommande övningar och den andra ska vi använda för att träna på omorganisation av sekundärt lagringsutrymme. Ett sätt att skapa två maskiner baserat på en `.OVA`-fil kan vara att göra *Import Appliance* och skapa en maskin och sedan byta namn på `.OVA`-filen och göra *Import Appliance* igen. Då får vi två olika maskiner, de kan kanske heta *NewTinyDebian* och *NewTinyDebian2*.

Starta nu *NewTinyDebian* med hjälp av *SystemRescueCD* (dvs montera CD:n och starta om maskinen). Med *gPartEd*, betrakta hur hårddisken ser ut. En enda stor partition och en liten swapdel. Inte bra. Skriv upp hur stora lagringsutrymmet var och en av katalogerna `/usr`, `/var`, `/tmp` och `/home` är. Notera också storleken på `swap`-partitionen. Man kan använda `df` och du för att få reda på storleken av partitioner och kataloger. Använd gärna option `-h` för att få mer human läslighet. (Använd `man` för att få mer information om hur `du` och `df` fungerar.)

Stäng av *NewTinyDebian* och, med hjälp av *VirtualBox*, skapa ytterligare en hårddisk som kopplas till *NewTinyDebian*. Låt dess storlek bli 6.0 GB. Den kommer troligtvis att dyka upp som `/dev/sdb` när du startar om *NewTinyDebian*. Starta återigen om datorn (boota på CD:n) och starta återigen *gPartEd*. Formatera den nya hårddisken och kopiera hela innehållet från `/dev/sda1` den gamla hårddisken till den nya. Du behöver inte kopiera `swap`-partitionen, bara `/dev/sda1`. OBS: Om du använder kommandot `cp` för att kopiera måste du se till att alla ägaregenskaper och rättigheter bevaras, det gör man med en speciell option till `cp`, läs manualsidan (`man cp`) för att veta vilken. Observera också att för att kopiera med `cp` måste man för det första montera partitionerna på två olika monteringspunkter (kataloger), dessa skapas lämpligen under `/mnt` med `mkdir`, själva monteringen sker med `mount`. Nu har du nytta av din förståelse av `mount` från förra övningen.

Ta nu bort alla filer ur katalogerna `/usr`, `/var`, `/tmp` och `/home` från `/dev/sda1`. (Med `rm -rf` på dessa kataloger, men försiktigt, kommandot `rm -rf` är farligt!). Ta också bort `swap`-partitionen. `Swap`-partitionen låg i en utvidgad partition som också kan tas bort, förminska `/dev/sda1` till några megabyte större än det som är kvar. Nu har du skapat den första partitionen i omorganisationen av *NewTinyDebian*. Den innehåller alla filer från `/` utom de som finns i katalogerna `/usr`, `/var`, `/tmp` och `/home`. Inför nu nya partitioner på detta sätt så att `/usr`, `/var`, `/tmp` och `/home` får varsinn egen partition. En bild på hur partitionsschemat kan se ut följer nedan.



Partition	File System	Size	Used	Unused	Flags
/dev/sda1	ext3	341.00 MiB	279.80 MiB	61.20 MiB	boot
▼ /dev/sda2	extended	2.17 GiB	--	--	
/dev/sda5	linux-swap	200.00 MiB	--	--	
/dev/sda6	ext2	1.27 GiB	1.17 GiB	96.82 MiB	
/dev/sda7	ext2	450.00 MiB	417.41 MiB	32.59 MiB	
/dev/sda8	ext2	150.00 MiB	6.26 MiB	143.74 MiB	
/dev/sda9	ext2	113.00 MiB	10.44 MiB	102.56 MiB	

Här är `/dev/sda7` farligt liten, gör den gärna lite större. Det är `/var` som ska monteras på den och om `/var`-partitionen blir full så kraschar datorn.

Vi ser att `swap`, `/usr`, `/var`, `/tmp` och `/home` ligger som logiska partitioner i en utökad partition. Här ser vi också slutresultatet då alla data är tillbakalagda i de respektive katalogerna.

När alla data är överflyttade behöver du skapa monteringspunkter i `/` (alltså på `/dev/sda1`) som de nya partitionerna ska monteras på vid systemstart och skriva om `/etc/fstab` för att reflektera de förändringar du gjort. Se till att `/usr` blir monterad som `read-only` vid start av systemet. Om allting gjorts korrekt ska systemet nu kunna startas om.

Ledning: Det här kan behövas göras om flera gånger, du kan börja om genom att göra *Import Appliance* igen och skapa en ny hårddisk igen etc. Glöm inte då att först ta bort den gamla maskinen som du försökte med.

Kommentar: Vi ser att i denna installation är `/boot` en del av den partition som `/` ligger på, detta är en skillnad mot vår tidigare *Gentoo*-installation.

Extra uppgift (överkurs): Kan du se till att systemet startar och även monterar `/` som `read-only`? Det finns någonting som heter `/etc/mtab` som påverkas av detta, hur ska det hanteras? Det är inte säkert att detta går, i skrivande stund har inte jag testat detta, men det kan vara värt att undersöka om det går, då lär vi oss mer om hur ett *POSIX*-system startar.

Extra fråga (överkurs): Om du lyckas med att montera både `/` och `/usr` i `read-only`, kan du montera `/var` och `/tmp` med `noexec`? Om du lyckas med det, har du då ett system som är helt skyddat från illvillig programvara som virus och liknande? Varför, varför inte?