

### Möte 3: Peer Instruction på installation och underhåll av operativsystem, samt lite TCP/IP

**Begreppstest 1:** Vid installation av operativsystem införs ett så kallat *installationsmedium*. Det brukar vara en körande variant av det operativsystem som ska installeras. Ett av följande påståenden är helt utan tvekan sant angående relationen mellan installationsmediet och det operativsystem som ska installeras. Vilket?

**A.** Om en drivrutin finns installerad i installationsmediet som möjliggör kommunikation med nätet så behöver man inte installera någon drivrutin för operativsystemet som ska installeras.

**B.** Eftersom installationsmediet kör så finns förstås en fungerande operativsystemskärna hörande till installationsmediet. Eftersom installationsmediet är en körande variant av operativsystemet som ska installeras så behöver man alltså inte installera en kärna för operativsystemet, man kan bara ta kärnan som hör till installationsmediet.

**C.** Filhierarkin som hör till installationsmediet innehåller filhierarkin som hör till det operativsystem som ska installeras.

**D.** Om installationsmediet används på ett redan färdiginstallerat system så är installationsmediet ett system som kör där användaren har administrationsrättigheter, alltså `root`-rättigheter. Det betyder dock inte att användaren som kör installationsmediet kan få `root`-rättigheter på den operativsystem som behandlas. Detta är en säkerhetsåtgärd för att inte vem som helst ska kunna få `root`-rättigheter på vilket system som helst. Då kan ju vem som helst bli `root` och det motverkar ju syftet med lösenord.

**Begreppstest 2:** När det gäller montering av hårddiskenheter (partitioner) så gäller följande:

**A.** Drivrutiner för det filsystem som man vill formatera behöver inte vara installerade om man inte vill förändra innehållet i enskilda filer, man kan *alltid* montera en partition som *read-only*, då behövs ingen hänsyn tas till vilket filsystem som partitionen är formaterad med.

**B.** Man behöver inte ta någon som helst hänsyn till formatering då det gäller montering av partitioner.

**C.** Montering av en partition kan göras endast då den är formaterad.

**D.** Inget av de ovanstående påståendena stämmer.

**Begreppstest 3:** Hierarkin i ett körande *POSIX*-system (som *GNU/Linux*, som vi arbetar med) har bland annat följande kataloger:

`/root, /usr, /etc, /proc, /dev`

I kataloger finner man filer och den *vanligaste* situationen är att:

- A.** `/root, /usr, /etc` är kataloger som innehåller filer på hårddisken, de andra är referenser till saker som inte nödvändigtvis direkt ligger på en hårddisk
- B.** `/root, /proc, /dev` är kataloger som innehåller filer på hårddisken, de andra är referenser till saker som inte nödvändigtvis direkt ligger på en hårddisk
- C.** `/usr, /etc, /proc` är kataloger som innehåller filer på hårddisken, de andra är referenser till saker som inte nödvändigtvis direkt ligger på en hårddisk
- D.** `/proc, /dev` är kataloger som innehåller filer på hårddisken, de andra är referenser till saker som inte nödvändigtvis direkt ligger på en hårddisk

**Begreppstest 4:** I ett *UNIX/POSIX*-system beskriver filen `/etc/fstab` hur partitionerna ska monteras vid start av systemet. Ett av följande påståenden är då sant. Vilket?

- A.** För att montera extra partitioner under drift av systemet måste `/etc/fstab` redigeras. För att vanliga användare ska kunna montera partitioner görs `/etc/fstab` skrivbar för alla.
- B.** Filen `/etc/fstab` kan beskriva monteringar av alla partitioner som systemet är i beröring med (kan detektera). Dock måste alla drivrutiner och systemverktyg finnas installerade för de filsystem som man avser att använda.
- C.** Om man som *vanlig användare* vill montera extra partitioner under drift av systemet kan man inte göra det med hjälp av att påverka filen `/etc/fstab`, man måste använda kommandot `mount` som är ett av de få kommandon som alla får köra alltid på alla möjliga sätt (inom vissa rimliga gränser).
- D.** Den enda användare som får hålla på med filen `/etc/fstab` är systemets administratör, det vill säga `root`-användaren. Det finns inget sätt att komma runt detta och det är av säkerhetsskäl.

Diskussion: Hur skapas `/etc/fstab` första gången?

*Egentligen finns det sanning i flera av dessa påståenden, det är kanske inte en ideal PI-fråga.*

**Begreppstest 5.** Vi antar att vi gör följande kommandon vid en *UNIX*-prompt:

```
$ ls -l
total 12
-rw-r--r-- 1 johnny johnny 6 May 16 11:25 fil1
-rw-r--r-- 1 johnny johnny 7 May 16 11:25 fil2
-rw-r--r-- 1 johnny johnny 8 May 16 11:25 fil3
$ mkdir dir1 dir2
$ mv fil1 dir1
$ mv -t dir2 fil2 fil3
$ mv dir1 fil1
$ mv dir2 fil2
$ ls -l
```

När vi nu gör `ls -l`, efter alla kommandon, vad ser vi då för resultat?

Här är ett utdrag ur manualsidan för `mv`:

```
NAME
    mv - move (rename) files

SYNOPSIS
    mv [OPTION]... [-T] SOURCE DEST
    mv [OPTION]... SOURCE... DIRECTORY
    mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

    Mandatory arguments to long options are mandatory for short options too.

    --backup[=CONTROL]
        make a backup of each existing destination file

    -t, --target-directory=DIRECTORY
        move all SOURCE arguments into DIRECTORY
```

- A. Vi ser två filer.
- B. Vi ser två kataloger.
- C. Vi ser återigen de filer som fanns i arbetskatalogen från början (`fil1`, `fil2` och `fil3`).
- D. Ingetdera av alternativen A-C är korrekta.

**Begreppstest 6.** Vi antar att vi kör virtuellt med katalogen `/media/sf_me` som delad katalog. Vi antar att vi gör följande kommandon vid en *UNIX*-prompt:

```
$ pwd
/home/me/test
$ ls -l
total 12
-rw-r--r-- 1 me me 6 May 16 11:25 fil1
-rw-r--r-- 1 me me 7 May 16 11:25 fil2
-rw-r--r-- 1 me me 8 May 16 11:25 fil3
$ mv fil1 /media/sf_me
$ cp fil2 /media/sf_me
$ cp fil3 /media/sf_me
$ rm fil3
```

Här är då alltså en katalog i värdsystemet monterad på `/media/sf_me` via *VirtualBox Guest Additions*.

Vilket påstående är sant?

A. Alla filer, `fil1`, `fil2` och `fil3` finns i separata kopior både i värdsystemet och den virtuella maskinen, om vi till exempel redigerar i `fil1` på värdsystemet så avspeglas förändringarna i den kopia som finns i den virtuella maskinen i `/home/me/test`.

B. Filen `fil2` finns i två separata kopior eftersom vi gjorde en `cp` på den, om vi ändrar i `/home/me/test/fil2` så avspeglas *inte* dessa förändringar i `/media/sf_me/fil2` eftersom det är två separata filer.

C. De två sista kommandona som utfördes,

```
$ cp fil3 /media/sf_me
$ rm fil3
```

har inte samma effekt på `fil3` som kommandot `mv fil1 /media/sf_me` har på `fil1`, filen `fil3` får nämligen ett nytt namn vid kopiering från `/home/me/test` till `/media/sf_me`, om man använder `mv` byts inte namnet på filen, den *bara* flyttas.

D. Ingetdera av alternativen A-C är sanna.