

1. Beskriv stegvis vilka datastrukturer och datablock måste accesseras för att läsa första blocket i en namngiven fil, om vi antar att inget finns i cache-minnet., samt att filen finns i root-katalogen i filsystemet. Dvs. börja med (3p)
  - a. För ett FAT-baserat system
  - b. För ett i-nod-baserat system

Modell:

För FAT

- I) Läs in block som representerar root-filsystem från skiva
- II) Traversera datastrukturen (katalogstrukturen) till filnamn hittas
- III) Läs ut på vilket block filen börjar
- IV) ....

För i-nod-baserat

....

Demo: Montera filsystem

- a) VFAT på Flash minnepinne
- b) ext3 som partition på hårddisk
- c) ISO som virtuell skiva (dvs .en fil representerande bitarna på skivan) via loop-enhet i Linux

2. Bygg ett program `mypasswd` för att kontrollera lösenord, med hjälp av unix funktionen `crypt()`. Systemet skall:
  - a. Spara namn lösenord i formen "username:encrypted\_password" i en textfil (välj själv namn)
  - b. alla lösenord skall matas in utan echo (dvs inte synas på skärmen)
  - c. kan checka password via `mypasswd <username>`
  - d. kan byta password via `mypasswd new <username>`(4p)

3. Anta att du genererar fotografier som kräver 10 GB lagringsutrymme per år. Bygg upp en strategi för att dina barnbarn skall kunna se på dina fotografier. Motivera din strategi, och påvisa också eventuella brister. (1p)
4. Skriv ett program som skapar (*fopen*, *fwrite*, *fclose*) 1000 filer med 1 byte i ett filsystem och sedan raderar (*remove*) dem. Mät tiden för operationerna. Prova följande
  - a. Filerna raderas i samma ordning som de skapas
  - b. Filerna raderas i motsatt ordning som de skapatsJämför tiderna. Spelar det någon roll i vilken ordning filerna raderas? (4p)