

OPERATIVSYSTEM 2015, ÖVNING 6, 16.12.2015

Lämnas in senast 15.1.2016, elektronisk på adressen <https://abacus.abo.fi/ro.nsf>

1. Med kommandot *mkisofs* kan man bygga upp ett ISO9660-filsystem (används typisk på CD-ROM / DVD). Ett exempel på ett sådant filsystem finns på

http://www.abo.fi/~jbjorkqv/opsys_12/test.iso

Ett ISO9660 filsystem består av sektorer på 2048 byte. De första 16 sektorerna (0-15) består av nollor (de är reserverade för framtida bruk). Sektor 17 innehåller *primary volume descriptor*, med format som i början på *struct t_isovoldesc* i koden nedan. Ladda ner *test.iso*, kompilera och testa koden. Lägg därefter till kod som dessutom skriver ut *sys_ident* samt totala antalet sektorer samt storleken på hela filsystem i bytes. Beskriv koden och bifoga resultat. (2p)

```
#include <stdio.h>
#define FILENAME "test.iso"
char volident[] = {67,68,48,48,49,1};

struct t_isovoldesc {
    char start;
    char desc[7];
    char sys_ident[32];
    char vol_ident[32];
    char zeros[8];
    long long sectors;
};

struct t_iso_sector {
    char data[2048];
};

int main() {
    FILE *fd;
    struct t_iso_sector iso_sector;
    struct t_isovoldesc *isovoldesc;

    fd = fopen(FILENAME, "r");
    while (!feof(fd)) {
        fread(&iso_sector, sizeof(iso_sector), 1, fd);
        isovoldesc = (struct t_isovoldesc *)&iso_sector;
        /* Search for sector containing volume descriptor */
        if (strncmp(isovoldesc->desc, volident,6)==0) break;
    }
    printf("Volume name: %s\n", isovoldesc->vol_ident);
    return 0;
}
```

2. Vi skall nu bygga ett ext4 filsystem. Först använder vi kommandot *dd* ("disk dump") för att skapa en tom fil

```
dd if=/dev/zero of=mydisk count=4096
```

dvs vi kopierar NULL-bytes till en fil med 1024 block. Sedan skapar vi ett filsystem av typen ext4 i denna fil

```
/sbin/mkfs.ext4 mydisk
```

sedan kan vi verifiera innehållet genom att köra en filsystemcheck.

```
/sbin/fsck.ext4 -f -v mydisk
```

Nästa steg skulle vara att montera filsystemet. Det går endast om man har root-access, så det får lämnas ogjort ifall man jobbar på tuxedo.

```
mount -o loop -o nosuid -o nodev mydisk mountpoint
```

Men hur som helst; när man skapade filsystemet, listades bla. antal i-noder. Var i filsystemet (i detta fall i filen "mydisk") sparas denna information?

Använd programmet **hexdump -C mydisk** för att se på bytes i filen. Tips; se på länken <http://myweb.lmu.edu/dondi/share/os/ext2-walkthrough.pdf>.

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

(4p)

4. Vi antar att det idag tar en timme att bryta krypteringssystem med nyckellängd på 56 bitar. Vilket år kan vi bryta ÅA:s nuvarande systems med 512 bitar nyckellängd på en timme, om vi antar att processeringskapaciteten fördubblas varje halvår? (2p)