OPERATIVSYSTEM 2008, RÄKNEÖVNING 4, v. 41, deadline må 13.10.2008. kl. 12

OBS Uppgift 5 måste besvaras!

1. En serieport tar emot en byte åt gången över en serielinje. Vid designen av drivrutinen övervägdes det att antingen använda aktivt väntande eller avbrottsbaserad läsning av inkommande data.
a) Om betjäning av avbrott (dvs. utföra avbrottsrutinen samt all extra tid som går åt till att aktivera avbrottsfunktionaliteten) tar 500 ns, vid vilken bit-rate (bit/sekund) lönar det sig att gå över från avbrottsbaserad till aktivt väntande? Vi kan negligera tiderna det tar att föra över data från serieporten till minnet.
b) Generellt använder sig hårdvaran ofta av en buffert med 16 bytes, för att kunna mellanlagra på serieporten inkommande data. Vilken effekt har det för valet mellan aktivt väntande och avbrottsbaserad läsning?
(2p)

2. Gör en undersökning av I/O-hårdvara på tuxedo.abo.fi (eller annan Linux maskin on du inte har tillgång till denna). Gör en rapport på följande saker: Vad finns det för hårdvara? Vad finns det för drivrutiner (device drivers), vilken typ är de (block- eller teckenbaserade). Hur många avbrott har drivrutinerna tagit emot? Vad kan man dra för slutsats av antalet avbrott (tänk på användningsmönstret)? Vad använder de för portar / minne? (2p)

TIPS: Se på ”filerna” /proc/devices, /proc/iomem, /proc/ioports, /proc/interrupts. Google kan också vara till nytta. Fråga någon du känner om hjälp att ”dekoda” vid behov.

3. Förfrågningar inkommer till hårdskivedrivrutinen för cylindrar 20, 10, 22, 2, 8, 40 och 28, i given ordning. En sökning tar 8 ms per cylinder man flyttar sig över. Hur lång total söktid behövs för

a) First come, first served
b) Närmaste cylindern till näst
c) Hissalgoritmen (till att börja med på väg uppåt)

Vi antar i samtliga fall att diskarmen till att börja med finns på cylinder 20. (2p)

4. Ett UNIX filsystem har 1 kB blockstorlek och 4 bytes disk-adresser. Vilken är max filstorlek, om i-noder innehåller 10 direkta referenser, en enkel indirektion, en dubbel resp, en trippel indirektion? (OBS vid indirektioner används blocken enbart för referenser). (2p)

5.Med kommandot *mkisofs* (finns på t.ex. lenz.cs.abo.fi) kan man bygga upp ett ISO9660-filsystem. Ett exempel på ett sådant filsystem finns på

http://www.abo.fi/~jbjorkqv/opsys_08/test.iso

Ett ISO9660 filsystem består av sektorer på 2048 byte varje. 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 (strcmp(isovoldesc->desc, volident)==0) break;
  }
  printf("Volume name: %s\n", isovoldesc->vol_ident);
  return 0;
}
```