Obligatorisk oppgave nr 3

INF2270

Frist 9. mai 2016 kl 10.00

Oppgaven er å programmere fire funksjoner

```
int  readbyte (FILE *f);
long readutf8char (FILE *f);
void writebyte (FILE *f, byte b);
void writeutf8char (FILE *f, unicode u);
```

som kan lese Unicode-tegn kodet som UTF-8 fra en fil og skrive UTF-8-kodete Unicode-tegn til en fil.

De to typene byte og unicode er definert slik:

```
typedef unsigned char byte;
typedef unsigned long unicode;
```

- Funksjonene skal programmeres i x86-assemblerkode slik at de fungerer med kommandoen gcc -m32 på Ifis Linux-maskiner.
- I mappen[1] ~inf2270/programmer/Oblig-3/ ligger filen oblig3-basis.s som kan være en god start til å skrive assemblerkoden.
- I samme mappe ligger filen test-oblig3.c som er det testprogrammet gruppelærerne vil bruke til å teste innlevert kode. Jeg vil anbefale deg også å bruke det.
- Ikke prøv å skrive all koden ferdig på én gang. Start med en kopi av basiskoden nevnt over og legg til litt av gangen; når du har testet den, kan du utvide med litt mer.
- Siden dette er en individuell oppgave, forventer jeg at alle besvarelsene er unike. Les Ifis regler om obligatoriske oppgaver og kopiering i http://www.uio.no/studier/admin/obligatoriske-aktiviteter/mn-ifi-oblig.html.

Del 1: writebyte

Denne funksjonen skrives slik i C:

```
void writebyte (FILE *f, byte b)
{
  fwrite(&b, 1, 1, f);
}
```

Den skriver én byte til filen overført som parameter.

Oversett denne funksjonen til assemblerkode.

Hint Datatypen FILE er definert i /usr/include/stdio.h, men du trenger ikke kjenne til denne. Du skal bare kopiere den pekeren som blir overført.

[1] Denne mappen kan man også få tilgang til over nettet i adressen http://inf2270.at.ifi.uio.no/programmer/Oblig-3/.

Hint Systemfunksjoner som fread og fwrite kalles på akkurat samme måte som de du har skrevet selv:

1. Legg parametrene på stakken (i omvendt rekkefølge).
2. Bruk instruksjonen call.
3. Fjern parametrene fra stakken.

Husk at de frie registrene **%EAX**, **%ECX** og **%EDX** kan bli endret.

Del 2: writeutf8char

Skriv denne funksjonen, som skriver ett Unicode-tegn kodet som UTF-8 til filen som er angitt som parameter ved å kalle på writebyte fra forrige del.

Hint Det lønner seg å gå gradvis frem: Skriv først en versjon som bare kan skrive ut 7-bits ASCII-tegn, deretter en som kan skrive ut Unicode-tegn opp til FFFF_{16}, osv.

Del 3: readbyte

Denne funksjonen skrives slik i C:

```
int readbyte (FILE *f)
{
  int status;
  char c;

  status = fread(&c, 1, 1, f);
  if (status <= 0) return -1;
  return (int)c;
}
```

Den leser én byte fra filen overført som parameter. Hvis det ikke er flere byte igjen på filen, skal funksjonen returnere -1.

Oversett denne funksjonen til assemblerkode.

Del 4: readutf8char

Skriv denne funksjonen som skal lese ett UTF-8-kodet Unicode-tegn fra filen ved å lese én eller flere byte ved å kalle på readbyte fra del 3. Hvis det ikke er flere tegn å lese (dvs at readbyte returnerte -1), skal readutf8char selv returnere -1.

Hint Du kan regne med at filen du leser fra alltid er korrekt kodet og kun ineholder lovlig UTF-8.