

Programmering i C/C++

Laboration 2 (21.9.2010)

Deadline för lösningar: **29.9.2010, kl. 23.59**

Lösningar till uppgifterna inlämnas elektroniskt på adressen: <https://xprog28.cs.abo.fi/ro.nsf>

OBS!! Denna laboration löses i C, dvs. INTE i C++!!

En löst uppgift utgörs av en dokumentation i PDF- eller Word-format. Denna dokumentation ska innehålla:

1. En introduktion till problemet och en beskrivning (i ord) hur den är löst
2. Listning av programkod
3. Skärmdump av utskrift från programkörning

Kontakta kursassistenten (stgronro@abo.fi) ifall du kör fast eller undrar över något!

Uppgift 1 (4p)

Skriv ett C-program som vid första körningen skriver en struktur med data in i en binär fil. Under följande körningar läses data från filen (allt på en gång) och innehållet skrivs ut på skärmen. Dvs. idén är att hantera situationen att filen inte finns (första körningen). Leta efter ett lämpligt felmeddelandevärde för globala variabeln `errno` i <errno.h>. Använd detta för att detektera felet med att filen inte existerar efter att fopen returnerar NULL.

Använd strukturen från föregående laboration, dvs:

```
typedef struct t_adress {
    char fornamn[MAX_NAMN_LANGD];
    char efternamn[MAX_NAMN_LANGD];
    char gatadress[MAX_ADRESS_LANGD];
    int postnummer;
    char postanstalt[MAX_POST_LANGD];
} kontakt;
```

Uppgift 2 (6p)

Fortsätt på programmet i uppgift 1. Skriv ut t.ex. 3 (olika) strukturer till den binära filen. Gör en funktion som läser in allt data i filen på en gång i en dynamiskt allokerad unsigned char buffer. Anta att du *inte* vet hur många strukturer som finns lagrade i filen när du läser den. Du kan t.ex. skriva en funktion som returnerar hur många adresser filen innehåller. En annan funktion skriver ut alla namn i databasen. Använd en pekare till en struct adress (kontakt *) för att skriva ut datat.

Exempel (men andra lösningar går också utmärkt) på prototyper till funktioner som kan implementeras:

```
int ge_antal_adresser(FILE *file); //Räknar ut antalet adresser i en fil
unsigned char *las_data_filen(FILE *file, int antal_strukturer); //Allokerar utrymme för och läser in
// antal_strukturer adresser från fil.
void skriv_ut_adresser(unsigned char *data_buffer, int antal_strukturer); //Skriv ut namnen
```

Kompilerera programmen så här: **gcc -Wall program.c -o program**

För ytterligare information angående kompileringen, skriv ”man gcc” i prompten.