

Lösning Repetitionsdugga 2022-05-31

Totalt 0.2 poäng, varav

- ✓ **0.1 p för korrekt svar på majoriteten av frågorna**
- ✓ **0.2 p för korrekt svar på samtliga frågor**

1. Du skall skapa en ny katalog och köra ett litet C-program i Linuxterminalen.

a) Från root, skapa en ny katalog döpt *dir2*. Dirigera till den nya katalogen och skapa en ny fil döpt *main.c*.

```
$ mkdir dir2  
$ cd dir2  
$ sudo nano main.c -l
```

b) I filen *main.c*, skriv ett minimalt program som skriver heltal 0 – 9 till textfilen *data.txt*.

```
/* Inkluderingsdirektiv: */  
#include <stdio.h>  
  
/*****  
 * main: Skriver heltal 0 - 9 till textfilen data.txt.  
 *****/  
int main(void)  
{  
    FILE* fstream = fopen("data.txt", "w");  
    if (!fstream) return 1;  
    for (register int i = 0; i < 10; ++i)  
        fprintf(fstream, "%d\n", i);  
    fclose(fstream);  
    return 0;  
}
```

c) Kompilera filen och skapa en körbar fil *main*. Kör sedan programmet.

```
$ gcc main.c -o main -Wall  
$ ./main
```

d) Kontrollera storleken på den körbara filen.

```
$ du -sh main
```

e) Radera hela mappen med alla filer i.

```
$ cd  
$ sudo rm -r dir2
```

2. I nedanstående exempel genomförs försök till utskrift till olika utenheter.

- a) Ange till vilken utenhet som respektive utskrift "Line 1" – "Line 7" kommer (eller inte kommer) skrivas till.

Se inline-kommentarer nedan.

- b) Programmet kompilerar, men kraschar vid körning. Förklara varför.

Det beror på att vi försöker genomföra utskrift med en nullpekare på rad 11 (0 = NULL), vilket inte fungerar.

```
1  int main(void)
2  {
3      FILE* fstream1 = fopen("data1.txt", "w");
4      FILE* fstream2 = fopen("data2.txt", "r");
5
6      fprintf(stdout, "Line 1!\n"); /* Konsolen. */
7      fprintf(fstream1, "Line 2!\n"); /* Filen data1.txt. */
8      fprintf(stdin, "Line 3!\n"); /* Ingenstans, stdin innebär läsning från konsolen. */
9      printf("Line 4!\n"); /* Konsolen (för printf används stdout).
10     fprintf(fstream2, "Line 5!\n"); /* Ingenstans, data2.txt öppnas enbart för läsning. */
11     fprintf(0, "Line 6!\n"); /* Ingenstans och programmet kraschar (utskrift med nullpekare). */
12     fprintf(stderr, "Line 7!\n"); /* Konsolen. */
13     return 0;
14 }
```

OBS! Vänd blad!

3. I följande program skall tio heltal tilldelas till ett dynamiskt fält fem gånger. Varje gång skrivs fältets innehåll ut både i konsolen samt till en textfil döpt *numbers.txt*. Programmet kompilerar dock inte. Markera samtliga fel och skriv korrigerade instruktioner på de rader fel detekteras.

Se korrigerad kod på nästa sida!

```

1  /* Inkluderingsdirektiv: */
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  /*****
6   * Returnerar pekare till dynamiskt allokerat fält av valbar storlek.
7   *****/
8  static inline int* intptr_new(const size_t* size) { return *(int*)malloc(sizeof(int) * size); }
9
10 /*****
11  * Frigör minne för dynamiskt allokerat fält.
12  *****/
13 static inline void intptr_delete(int** self) { free(*self); *self = 0; }
14
15 /*****
16  * Fyller en array till bredden med jämna tal. Vid varje anrop används olika tal
17  * för tilldelningen via användning av en statisk variabel döpt num, som
18  * inkrementeras efter varje tilldelning.
19  *****/
20 static void assign(int* data, const size_t size)
21 {
22     static int* num = 0;
23     for (register int* i = *data; i < *data + size; ++i)
24         i = 2 * *num++;
25     return;
26 }
27
28 /*****
29  * Bifogar heltal lagrade i ett fält till en fil.
30  *****/
31 static void print(const int* data, const size_t size, FILE* stream)
32 {
33     if (!stream) *stream = *stdout;
34     fprintf(stream, "-----\n");
35     for (register const int* i = *data; i < *data + size; ++i)
36         fprintf(stream, "%d\n", i);
37     fprintf(stream, "-----\n\n");
38     return;
39 }
40
41 /*****
42  * Fyller dynamisk array med fem kombinationer av tio jämna heltal. Varje kombination
43  * skrivs ut i konsolen och bifogas till filen numbers.txt.
44  *****/
45 int main(void)
46 {
47     int* data = intptr_new(10);
48     FILE fstream = *fopen("numbers.txt", "a");
49
50     for (register time_t i = 0; i < 5; ++i)
51     {
52         assign(data, 10);
53         print(data, 10, stdout);
54         print(data, 10, &fstream);
55     }
56
57     intptr_delete(&data);
58     fclose(&fstream);
59     return 0;
60 }

```

Korrigerad kod till uppgift 3:

```

/* Inkluderingsdirektiv: */
#include <stdio.h>
#include <stdlib.h>

/*****
 * Returnerar pekare till dynamiskt allokerat fält av valbar storlek.
 *****/
static inline int* intptr_new(const size_t size) { return (int*)malloc(sizeof(int) * size); }

/*****
 * Frigör minne för dynamiskt allokerat fält.
 *****/
static inline void intptr_delete(int** self)
{
    free(*self);
    *self = 0;
    return;
}

/*****
 * Fyller en array till bredden med jämna tal. Vid varje anrop används olika tal
 * för tilldelningen via användning av en statisk variabel döpt num, som
 * inkrementeras efter varje tilldelning.
 *****/
static void assign(int* data, const size_t size)
{
    static int num = 0;
    for (register int* i = data; i < data + size; ++i)
        *i = 2 * num++;
    return;
}

/*****
 * Bifogar heltal lagrade i ett fält till en fil.
 *****/
static void print(const int* data, const size_t size, FILE* stream)
{
    if (!stream) stream = stdout;
    fprintf(stream, "-----\n");
    for (register const int* i = data; i < data + size; ++i)
        fprintf(stream, "%d\n", *i);
    fprintf(stream, "-----\n\n");
    return;
}

/*****
 * Fyller dynamisk array med fem kombinationer av tio jämna heltal. Varje kombination
 * skrivs ut i konsolen och bifogas till filen numbers.txt.
 *****/
int main(void)
{
    int* data = intptr_new(10);
    FILE* fstream = fopen("numbers.txt", "a");

    for (register size_t i = 0; i < 5; ++i)
    {
        assign(data, 10);
        print(data, 10, stdout);
        print(data, 10, fstream);
    }

    intptr_delete(&data);
    fclose(fstream);
    return 0;
}

```