

Lösningar - Dugga 2022-06-07

1. Du har fyra lysdioder anslutna till PIN 17, 22, 23 samt 24, som skall blinkas i en sekvens via ett shell-script döpt *main.sh*. I denna uppgift skall Linuxkommandon för att skapa/radera kataloger och filer samt innehållet i shell-scriptet anges.

- a) Förklara varför man brukar skriva *#!/bin/bash* längst upp i ett shell-script.

Säger åt operativsystemet att använda Bash som interpretator.

- b) Skapa en ny katalog döpt *bash*. I denna katalog, skapa en ny barnkatalog/underkatalog döpt *ledsequence1*, dirigera till denna och skapa ett nytt shell-script döpt *main.sh* utan att öppna denna. Lista sedan samtliga filer i katalogen för att verifiera att shell-scriptet har skapats. Öppna slutligen shell-scriptet *main.sh* med radnummer.

```
$ mkdir bash
$ cd bash
$ mkdir leqsequence1
$ cd ledsequence1
$ touch main.sh
$ ls
$ sudo nano main.sh -l
```

- c) Skriv kommandon i shell-scriptet *main.sh* så att lysdioderna blinkar i en sekvens en gång varannan sekund när scriptet körs. Använd det virtuella filsystemet *sysfs*. Lysdioderna skall blinka i följande ordning för en enda sekvens: 17 => 22 => 23 => 24 => 23 => 22. Lysdioderna skall alltså blinka framåt och sedan bakåt.

```
1  #!/bin/bash
2
3  # Reserverar PINS:
4  echo 17 > /sys/class/gpio/export
5  echo 22 > /sys/class/gpio/export
6  echo 23 > /sys/class/gpio/export
7  echo 24 > /sys/class/gpio/export
8
9  # Sätter samtliga PINS till utportar:
10 echo out > /sys/class/gpio/gpio17/direction
11 echo out > /sys/class/gpio/gpio22/direction
12 echo out > /sys/class/gpio/gpio23/direction
13 echo out > /sys/class/gpio/gpio24/direction
14
15 # Blinkar LEDs kontinuerligt i en sekvens varannan sekund::
16 while true
17 do
18     echo 1 > /sys/class/gpio/gpio17/value
19     sleep 2
20
21     echo 0 > /sys/class/gpio/gpio17/value
22     echo 1 > /sys/class/gpio/gpio22/value
23     sleep 2
24
25     echo 0 > /sys/class/gpio/gpio22/value
26     echo 1 > /sys/class/gpio/gpio23/value
27     sleep 2
28
29     echo 0 > /sys/class/gpio/gpio23/value
30     echo 1 > /sys/class/gpio/gpio24/value
31     sleep 2
32
33     echo 0 > /sys/class/gpio/gpio24/value
34     echo 1 > /sys/class/gpio/gpio23/value
35     sleep 2
```

Hårdvarunära programmering

```
36 echo 0 > /sys/class/gpio/gpio23/value
37 echo 1 > /sys/class/gpio/gpio22/value
38 sleep 2
39
40 echo 0 > /sys/class/gpio/gpio22/value
41 done
```

- d) Kör shell-scriptet *main.sh*. Dirigera sedan till förälderkatalogen *bash* och radera sedan barnkatalogen *ledsequence1*. Lista samtliga filer för att verifiera att katalogen är raderad.

```
$ sudo bash main.sh
$ cd ..
$ rm -r ledsequence1
$ ls
```

2. I nedanstående C-program används det virtuella filsystemet *sysfs* för att tända en lysdiod ansluten till PIN 17 vid nedtryckning av en tryckknapp ansluten till PIN 27, övrig tid skall lysdioden hållas släckt. Programmet kompilerar dock inte.

- a) Kontrollera programmet, lokalisera eventuella fel, kryssa för dessa och skriv korrigerande instruktioner vid sidan av.

Se lösning längst ned i detta dokument!

- b) Ange Linuxkommandon för att i din katalog *sysfs* skapa en ny katalog döpt *error_code1*. I denna katalog skapa en ny fil döpt *main.c*. Antag att programmet klistras in i denna fil. Kompilera sedan filen *main.c* med GCC-kompilatorn och skapa en körbar fil döpt *ledblink*. Kör sedan programmet.

```
$ cd sysfs
$ mkdir error_code1
$ cd error_code1
$ nano main.c -l
$ gcc main.c -o ledblink -Wall
$ ./main
```

Felkod - Filen *main.c*:

```
1  /* Inkluderingsdirektiv: */
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  /*****
6  * file_write: Skriver en rad med text till en fil.
7  *****/
8  static void file_write(const char* filepath, const char s)
9  {
10     FILE fstream = *fopen(filepath, "w");
11
12     if (fstream)
13     {
14         fprintf(stderr, "Could not open file at path %s!\n\n", filepath);
15     }
16     else
17     {
18         fprintf(fstream, "%d", s);
19     }
20     return;
21 }
```

Hårdvarunära programmering

```
22 /*****
23 * file_read: Läser en rad med text från en fil och lagrar i en sträng.
24 *****/
25 static void file_read(const char* filepath, const char s, const size_t size)
26 {
27     FILE fstream = *fopen(filepath, "r");
28     if (fstream)
29     {
30         fprintf(stderr, "Could not open file at path %s!\n\n", *filepath);
31     }
32     else
33     {
34         fgets(s, (int)size, fstream);
35     }
36     return;
37 }
38
39
40 /*****
41 * main: Ansluter en lysdiod till PIN 17 samt en tryckknapp till PIN 27. Angivna
42 *       PINs reserveras för aktuell process, följt av att datariktning väljs.
43 *       Vid nedtryckning av tryckknappen tänds lysdioden, annars hålls den släckt.
44 *       Tryckknappens insignal läses via en sträng s, vars innehåll efter läsning
45 *       typomvandlas till int via funktionen atoi från biblioteket stdlib.h.
46 *       Ifall tryckknappen blir heltalet 1, annars 0.
47 *****/
48 main()
49 {
50     char* s[10];
51     s[0] = '\0';
52
53     file_write("sys/class/gpio/export", "17");
54     file_write("sys/class/gpio/export", "27");
55     file_write("sys/class/gpio/17/direction", "out");
56     file_write("sys/class/gpio/27/direction", "in");
57
58     for (;;)
59     {
60         file_read("sys/class/gpio/27/value", *s, sizeof(*s));
61
62         if (*atoi(s) == 1)
63         {
64             file_write("sys/class/gpio/17/value", "1");
65         }
66         else
67         {
68             file_write("sys/class/gpio/17/value", "0");
69         }
70     }
71
72     return;
73 }
```

Lösning - Filen *main.c*:

Kontrollerar första inlästa tecknet från filen */sys/class/gpio/gpio27/value* i stället för att typomvandla detta till ett heltal via anrop av funktionen *atoi*. Därmed kan biblioteket *stdlib.h* utelämnas.

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

/*****
 * file_write: Skriver en rad med text till en fil.
 *****/
static void file_write(const char* filepath, const char* s)
{
    FILE* fstream = fopen(filepath, "w");

    if (!fstream)
    {
        fprintf(stderr, "Could not open file at path %s!\n\n", filepath);
    }
    else
    {
        fprintf(fstream, "%s", s);
        fclose(fstream);
    }

    return;
}

/*****
 * file_read: Läser en rad med text från en fil och lagrar i en sträng.
 *****/
static void file_read(const char* filepath, char* s, const size_t size)
{
    FILE* fstream = fopen(filepath, "r");

    if (!fstream)
    {
        fprintf(stderr, "Could not open file at path %s!\n\n", filepath);
    }
    else
    {
        fgets(s, (int)size, fstream);
        fclose(fstream);
    }

    return;
}
```

Hårdvarunära programmering

```
/******  
* main: Ansluter en lysdiod till PIN 17 samt en tryckknapp till PIN 27. Angivna  
*       PINS reserveras för aktuell process, följt av att datariktning väljs.  
*       Vid nedtryckning av tryckknappen tänds lysdioden, annars hålls den släckt.  
*****/  
int main(void)  
{  
    char s[10];  
    s[0] = '\0';  
  
    file_write("/sys/class/gpio/export", "17");  
    file_write("/sys/class/gpio/export", "27");  
    file_write("/sys/class/gpio/gpio17/direction", "out");  
    file_write("/sys/class/gpio/gpio27/direction", "in");  
  
    while (1)  
    {  
        file_read("/sys/class/gpio/gpio27/value", s, sizeof(s));  
  
        if (s[0] == '1')  
        {  
            file_write("/sys/class/gpio/gpio17/value", "1");  
        }  
        else  
        {  
            file_write("/sys/class/gpio/gpio17/value", "0");  
        }  
    }  
  
    return 0;  
}
```