TECHNISCHE UNIVERSITÄT
CHEMNITZ

Professur
Betriebssysteme

4. Übungsblatt

Grundlagen der Anwendungsunterstützung

Christine Jakobs und Laura Morgenstern

Aufgabe 1

Bringen Sie folgendes C-Programm zur Ausführung. Dokumentieren Sie den Vorgang von der Erstellung der Quellcode-Datei bis zur Programmausführung. Nutzen Sie den Compiler GCC.

```
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
  int antwort;
  int x = 2;

  antwort = x * (x + 1) * 7;
  printf("Die␣Antwort␣lautet␣%d\n", antwort);

  return antwort;
}
```

Aufgabe 2 (Zusatzaufagbe)

Folgendes C-Programm werde in die ausführbare Datei `program` übersetzt.

```
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
  unsigned int i;

  printf("%d\n", argc);
  for(i = 0; i < argc; i++)
    printf("%d:␣%s\n", i, argv[i]);

  return 0;
}
```

a) Welche Ausgabe erscheint auf dem Bildschirm, wenn es über die Kommandozeile mit folgenden Kommandos aufgerufen wird?

(i) `./program`
(ii) `./program Hallo Welt`
(iii) `./program "Hallo Welt"`

b) Wie gelangen die Daten über `argc` und `argv` ins Programm?

Aufgabe 3 (Zusatzaufgabe)

Informieren Sie sich in den Manpages [1] über die Systemrufe `fork()`, `wait()`, `exit()` (Section 2).

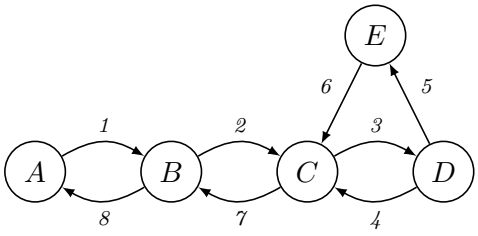
Aufgabe 4

Welche Ausgabe erscheint auf dem Bildschirm, wenn folgendes Programm gestartet wird? Wie ist dieses „ungewöhnliche“ Verhalten erklärbar?

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
  pid_t p;

  p = fork();
  if(p == 0)
    printf("Null\n");
  else
    printf("nicht␣Null\n");

  return 0;
}
```

Aufgabe 5

Betrachten Sie das klassische Prozesszustandsmodell.



a) Welche Zustände beschreiben die einzelnen Knoten?

b) Welche Systemrufe oder Ereignisse bewirken einen Übergang zwischen den Zuständen. Welche Aktionen müssen ggf. bei einem Zustandsübergang ausgelöst werden?

Aufgabe 6

Gegeben sei folgende Codesequenz. Erstellen Sie für dieses Programm ein Ablaufdiagramm! Welche Prozesse befinden sich zu den Zeitpunkten $1\,\mathrm{s}, 3\,\mathrm{s}, \ldots, 11\,\mathrm{s}$ in der `sleep()`-Funktion?

```
int main()
{
  pid_t p;

  sleep(2);
  fork();
  sleep(2);
  p = fork();
  if(p > 0)
    wait(NULL);
  sleep(2);
  fork();
  sleep(2);
  exit(0);

  return 0;
}
```

Was fällt Ihnen bei genauerer Betrachtung des Codes auf?

Literatur

[1] `http://linux.die.net/man/`, Online-Sammlung verschiedener Linux-Manpages