

Systemmanagement und Sicherheit

4. Übung

Aufgabe 1 (fork-exec)

Entwickeln Sie ein C-Programm *start*, das beim Aufruf

```
start prog arg1 arg2 arg3 ...
```

zunächst ein *fork()* Aufruf ausführt und dann im Sohnprozeß das Programm *prog* via *execvp* mit den angegebenen Argumenten startet.

Der Vaterprozess soll weiterhin folgendes tun:

- Ausgabe der PID des gestarteten Prozesses *prog*,
- Ausgabe des Return-Codes von *prog* nach dessen Beendigung,
Hinweis: Siehe Macros unter *wait(2)*
- Ausgabe eines evtl. Signals (numerisch, falls möglich auch symbolisch), das zum Abbruch von *prog* führte.
Hinweis: Das Signal mit der Nummer 10 senden Sie an einen Prozeß mit PID 4711 mit Hilfe des Kommandos

```
kill -10 4711
```

Fragen zum Thema:

- a) Welche Fehlercodes können beim Aufruf von *fork()* auftreten?
- b) Wie erhält man die PID von *prog*?
- c) Die Deklaration von *main()* in *start.c* sei

```
int main(int argc, char **argv)
```

Welcher Aufruf von *execvp()* ist dann richtig?

- i) `execvp(argv[1][0], argv[1])`
- ii) `execvp(argc, argv)`
- iii) `execvp(argv[1], argv[2])`
- iv) `execvp(*argv[1], *argv[1])`
- v) `execvp(**argv, **argv[1])`
- vi) `execvp(argv[1], argv[1])`
- vii) `execvp(argv[1], argv+1)`