Übungen zu Systemnahe Programmierung in C

Abschnitt 13.3: Aufgabe (mish - Teil B & C)

20.07.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Aufgabe: mish - Teil b)

Signalbehandlung von SIGINT

- Anpassen der Signalbehandlungen für CTRL+C
- SIGINT wird allen Prozessen des Terminals zugestellt

```
01 $> ./mish
02 mish> sleep 2
03 Exit status [5321] = 0
04 mish> sleep 10000
05 ^C                          # CTRL+C
06 $>
```

⇒ Bei CTRL+C stirbt `sleep` und `mish`

- Anpassen der Signalbehandlung:
  - Vater: Signal ignorieren (SIG_IGN)
  - Kind: Default-Behandlung (SIG_DFL)

Aufgabe: mish - Teil b)

Aufsammeln von Zombieprozessen

- Bisher: Aufsammeln durch `waitpid(2)` (blockierend)
- Signal `SIGCHLD` zeigt Statusänderung von Kindprozessen an
  - Kindprozess wurde gestoppt
  - Kindprozess ist terminiert
- Jetzt: Aufsammeln durch `waitpid(2)` (nicht-blockierend)
- Warten auf Statusveränderungen mit `sigsuspend(2)`

Aufgabe: mish - Teil c)

Unterstützung von Hintergrundprozessen

- Kommandos mit abschließenden '&'

  ⇒ Hintergrundprozess
- Beispiel: `./sleep 10 &`
- Ausgabe der Prozess-ID und des Prompts
- Anschließend sofort Entgegennahme neuer Befehle

```
01 # Starten eines Hintergrundprozesses mit &
02 mish> sleep 10 &
03 Started [2110]
04 mish> ls
05 Makefile mish mish.c
06 Exit Status [2115] = 0
07 ...
08 Exit status [2110] = 0
```

Aufgabe: mish - Teil c)

Unterstützung von Hintergrundprozessen

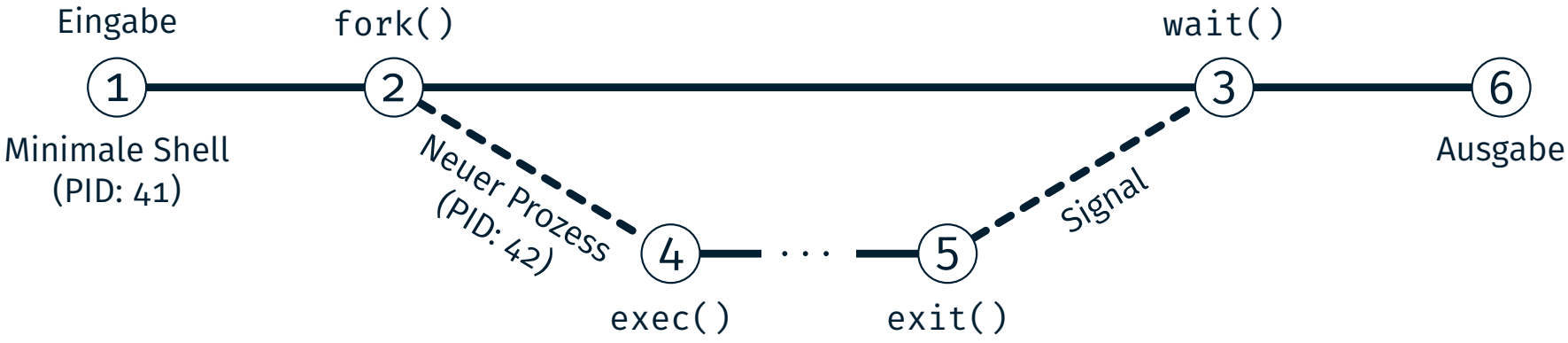
- Beim Warten auf Vordergrundprozesse sollen terminierende Hintergrundprozesse sofort eingesammelt werden

```
01 # Starten mehrerer Hintergrundprozesse
02 mish> sleep 3  &
03 Started [2110]
04 mish> sleep 5  &
05 Started [2115]
06 mish> sleep 10 &
07 Started [2118]
08
09 # Starten eines Vordergrundprozesses
10 mish> sleep 20
11 Exit Status [2110] = 0    # sleep 3  &
12 Exit Status [2115] = 0    # sleep 5  &
13 Exit Status [2118] = 0    # sleep 10 &
14 Exit Status [2121] = 0    # sleep 20
15 mish>
```

Aufgabe: mish - Teil c)

- Erweiterung des Basisablaufs

1. Auf Eingaben vom Benutzer warten
2. Neuen Prozess erzeugen
3. Vater: Wartet auf die Beendigung des Kindes
4. Kind: Startet Programm
5. Kind: Programm terminiert
6. Vater: Ausgabe der Kindzustands



Aufgabe: mish - Teil c)

- Erweiterung des Basisablaufs

1. Auf Eingaben vom Benutzer warten
2. Neuen Prozess erzeugen
3. Vater: Wartet auf die Beendigung des Kindes *(Nur Vordergrund)*
4. Kind: Startet Programm
5. Kind: Programm terminiert
6. Vater: Ausgabe der Kindzustands