



Prozesse

letzte Änderung: 6. Mai 2020

1. Aufgabe: Verwaister Kindprozess

Abgabedatei(en): `orphan.c`

„Beendet sich der Elternprozess vor dem Kindprozess, dann adoptiert der Init Prozess (PID 1) den Kindprozess und wird zu seinem neuen Elternprozess.“

1. Schreiben Sie eine Funktion

```
void orphan()
```

um diese Aussage zu verifizieren.

2. Erweitern Sie den Elternprozess um einen Exithandler, der den String "Goodbye" gefolgt vom Namen des Prozesseigentümers ausgibt.

2. Aufgabe: Zombie, Zombie, ZOMBIE!!

Abgabedatei(en): `zombie.c`

Schreiben Sie eine Funktion

```
void zombie(int argc, char ** argv)
```

welche k Zombieprozesse generiert. Die Anzahl k soll als Kommandozeilenparameter übergeben werden. Überprüfen Sie mit dem Kommando `ps` den Erfolg Ihres Programms.

Für $k = -1$ soll das Programm soviel Zombieprozesse wie möglich generieren. Versuchen Sie, ein reguläres Programm zu öffnen sobald Ihr Programm keine weiteren Zombieprozesse mehr generieren kann. Wie erklären Sie sich das Verhalten des Systems (Antwort als Kommentar im Quellcode des Hauptprogramms abgeben)?

3. Aufgabe: Prozess Environment

Abgabedatei(en): `environment.c`

Erstellen Sie eine Funktion

```
void environment()
```

die interaktiv das Ändern bzw. Erfragen der Environment-Variablen ermöglicht.

```
-----  
Environment-List Menu  
-----
```

```
[1]: Print environment list  
[2]: Print environment entry (getenv)  
[3]: Add environment entry (putenv)
```



```
[4]: Modify environment entry (setenv)
[5]: Remove environment entry (unsetenv)
[0]: End
```

Hinweis: Verwenden Sie die Funktion `readline()` oder `getline` um Benutzereingaben einzulesen.
Dazu verwenden Sie die Bibliothek *libreadline*.