

Job und Prozesskontrolle:

Jobs:

Auf der Shell gibt es die Möglichkeit mehrere Programme gleichzeitig auszuführen. Jedoch kann immer nur ein Programm angezeigt werden. Deshalb gibt es die Möglichkeit Programme im Hinter- und im Vordergrund zu starten.

Um ein Programm im Hintergrund zu starten, wird ein „&“ hinter das Programm geschrieben. z.B.:

xeyes &

So kann man jeden Befehl im Hintergrund starten, jedoch sollte der Befehl weder eine Ausgabe erzeugen, noch eine Eingabe erwarten, wenn doch, sollten diese umgeleitet werden z.B.

programm <Eingabedatei >Ausgabedatei 2>&1

Wurde ein Programm im Vordergrund gestartet, kann es durch die Tastenkombination **<Strg> + <z>** gestoppt werden. Dann kann diese Programm durch den Befehl:

fg <jobnummer>

im Vordergrund fortgesetzt, oder durch:

bg <jobnummer>

im Hintergrund fortgesetzt werden.

So können nun mehrere Programme im Hintergrund laufen. Um hier den Überblick zu behalten, gibt es das Programm:

jobs

dessen Ausgabe folgendermaßen aussehen könnte:

[1]	Running	xeyes &
[2]-	Running	xclock &
[3]+	Stopped	xeyes

Hier ist nun die Job-Nummer, der Jobs-Status und der Job-Name zu sehen.

Um nun einen Job zu beenden ohne ihn erst in den Vordergrund zu holen und ihn dann mit **<Strg> + <c>** zu beenden, kann der Befehl:

kill %<jobnummer>

verwendet werden. Um z.B. das Programm xclock zu beenden.

kill %2

Prozesskontrolle:

Unter Linux hat jedes laufende Programm eine Prozessnummer. Mit dem Befehl **ps** kann man sich alle laufenden Programme ansehen.

Mit **ps -ef** kann man sich alle Programme (**e**) mit allen Informationen (**f**) anzeigen lassen. Diese Ausgabe könnte folgendermaßen aussehen:

```
UID          PID  PPID  C  STIME TTY          TIME CMD
root          1      0  0  19:01 ?            00:00:00 init [3]
root          2      0  0  19:01 ?            00:00:00 [kthreadd]
m_hartig    5480  5479  0  19:35 pts/1        00:00:00 /bin/bash
m_hartig    5846  5480  0  20:40 pts/1        00:00:00 ps -ef
```

Hier sehen wir die UID, also wem das Programm gehört, die PID, die sogenannte Prozess-ID, die PPID, also die Prozess-ID des Eltern-Prozesses. Den Status eines Programms „C“, die STIME also die Startzeit des Programms, TTY der Ort auf dem das Programm läuft, TIME die verbrauchte CPU-Zeit und CMD der Programmname.

Programme die in eckigen Klammern angezeigt werden, sind Kernel-Programme.

Allen Programmen kann man mit dem Befehl **kill** Signale senden. Eine Liste aller Signale bekommt man mit **kill -l**.

Hier sind die Wichtigsten:

```
1  SUGHUP          9  SIGKILL          15  SIGTERM          19  SIGSTOP
18  SIGCONT
```

Wird **kill** ohne Option aufgerufen, wird das Signal 15 versendet. Die Signale 9 und 19 sind die Einzigen gegen die sich kein Befehl wahren kann. Alle anderen können von jedem Programm individuell verarbeitet werden.

Signal 1 bedeutet ein Neustarten des Programms, 15 bedeutet das Beenden des Programms auf normale Art, 9 bedeutet sofortiges beenden des Programms ohne Widerrede, 19 ist das Anhalten eines Programmes und 18 das Fortsetzen.

Um nun das Signal 9 an das Programm „/bin/bash“ mit der PID 5480 zu senden:

```
kill -KILL 5480
```

oder

```
kill -9 5480
```

eingeben.

Wird ein Eltern-Programm beendet, werden auch alle Kind-Programme beendet, es sei denn, sie werden mit dem Programm **nohup** gestartet. z.B.:

```
nohup xeyes
```

Prozessprioritäten:

Unter Linux, können jedem Prozess unterschiedliche Prioritäten zugewiesen werden. Somit kann bestimmt werden, wieviel CPU Leistung ein Prozess für sich beanspruchen darf.

Das sind die Sogenannten Nice-Level. Es gibt die Nice-Level von -20 bis +19, wobei ein niedrigerer Nice-Level eine höhere Priorität bedeutet. Root kann alle Nice-Level vergeben, ein normaler Benutzer kann nur Positive Nice-Level verwenden. Standardmäßig werden alle Prozesse mit dem Nice-Level 0 gestartet.

Soll ein Prozess mit einem anderen Nice-Level gestartet werden, wird das Programm **nice** vor dem eigentlichen Programm aufgerufen.

nice -n 5 xeyes

Startet z.B. das Programm xeyes mit einem Nice-Level von 5

nice -n -7 xeyes

Startet das Programm xeyes mit einem Nice-Level von -7

Man kann auch den Nice-Level bereits laufender Programme ändern. Root kann wieder beliebige Nice-Level vergeben, ein normaler Benutzer kann die Nice-Level jedoch nur erhöhen, also die Priorität senken.

renice 6 5436

Ändert den Nice-Level des Programms mit der PID 5436 auf 6

renice -4 5436

Ändert den Nice-Level des Programms mit der PID 5436 auf -4