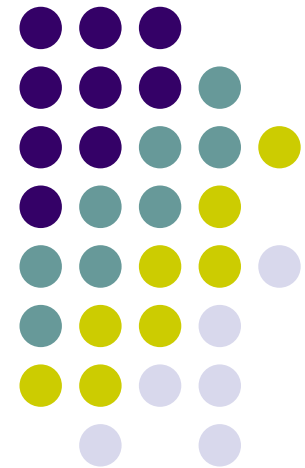


Unixkurs

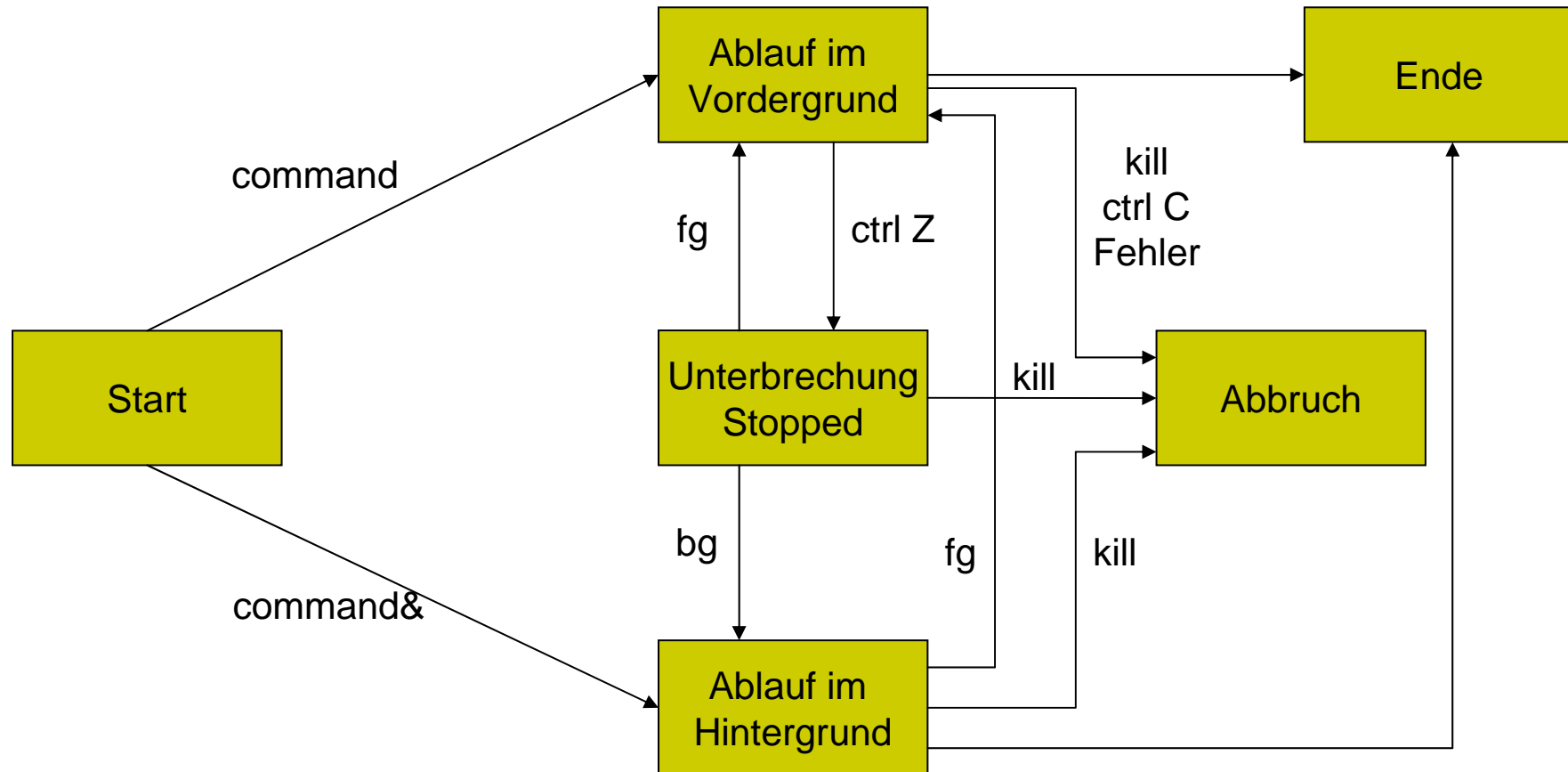
Prozesse
Editoren
Erstellen von C-Programmen



Prozesse

- Jedesmal wenn ein Programm gestartet wird , wird in Unix ein **neuer Prozeß erzeugt**.
- Zur Steuerung der Prozesse können Anwender Tastenkürzel und Kommandos benutzen.

Prozessmanipulation



Prozessmanipulation 2

- Zum Starten eines Prozesses **im Hintergrund** muss an das Kommando ein **&** angehängt werden.
- Beim Start des Prozesses im Hintergrund wird die **Jobnummer** und die **Prozessidentifikation (PID)** ausgegeben.
- Zum **Abbrechen** eines Prozesses kann das **kill** Kommando benutzt werden.
- Das Kommando akzeptiert entweder die **Jobnummer** oder die **Prozessidentifikation** als Argument.
- Bei der Identifizierung über die Jobnummer muss ein Prozentzeichen (%) vor der Jobnummer angegeben werden.

Signale

- kill
 - -1: SIGHUP (lese Konfigurationsdateien neu)
 - -2: SIGINT (interrupted)
 - -3: SIGQUIT (quit)
 - -9: SIGKILL (kann nie abgefangen werden)
 - -15: SIGTERM (voreingestellt)
- killall: [-Signalnummer] Prozeßname
 - Signale an alle Prozesse gleichen Namens (z. B. xclock)

Prozessmanipulation 3

- Zum **Abbrechen** eines Jobs im **Vordergrund** wird benutzt:
 - `ctrl^c` (Cancel)
- Zustand, Status und Jobnummer der Jobs kann gelistet werden durch
 - `jobs`
- Zum **Stoppen** eine Jobs dient
 - `Ctrl^Z`.
- ein Vordergrundjob kann in den Hintergrund geschickt werden mit
 - `bg` (background)
- Ein Hintergrundjob kann in den Vordergrund geholt werden mit
 - `fg` (foreground)

Prozessanzeige

Das Kommando `ps` (`process status`) zeigt die Prozesse mit ihrem Status an.

- Die Prozeßtabelle wird mit einer Titelzeile ausgegeben.
- Die wichtigsten Spalten bedeuten folgendes:
 - `COMMAND` der Name des Kommandos; Prozesse, die komplett in den Swapbereich ausgelagert sind, werden in Klammern angezeigt.
 - `TIME` die verbrauchte Rechenzeit (Summe User- und Kernelmodus) im Format MM:SS
 - `TT` die Nummer des kontrollierenden Terminals

Prozeßanzeige 2

- **UID** die Benutzer-ID des Eigentümer dieses Prozesses
- **PID** die Prozessnummer dieses Prozesses
- **PPID** die Prozessnummer des Elternprozesses
- **PGID** die Prozessgruppe dieses Prozesses
- **TPGID** die Prozessgruppe, der z.Z. das kontrollierende Terminal zu diesem Prozess gehört
- **SID** die Session-ID des Prozesses (ID der Loginshell)

Prozessanzeige 3

- **STAT** ist der Status des Prozesses; folgende Symbole sind möglich:
 - R (Running) - lauffähig
 - S (Sleeping) - schlafend
 - D (Dormant) - nicht störbarer Schlaf
 - T (sTopped) - angehalten
 - Z Zombie
- **F** die Flags des Prozesses
- **PRI** das aktuelle Maximum an Rechenzeit (in Millisekunden), das dem Prozeß zugeteilt wird, wenn der Prozess gerade läuft
- **NICE** ist der Nicewert des Prozesses und zeigt an, wie die Priorität des Prozesses erniedrigt wurde gegenüber anderen Prozessen
- **WCHAN** ist der Name der Kernelfunktion, in der der Prozeß schläft

Die wichtigsten Optionen des ps-Kommandos sind:

- -a zeigt die Prozesse aller User
- -c zeigt den Namen des Kommandos
- -e zeigt die Prozeßumgebung
- -h& unterdrückt die Kopfzeile
- -j jobs Format: PGID und SID
- -l langes Format (FLAGS WCHAN NICE PRIO)
- -m zeigt Speichernutzung
- -X zeigt EIP (Extended Instruction Pointer) und ESP (Extended Stack Pointer)
TIMEOUT ALARM
- -n gibt numerische Werte für USER und WCHAN
- -r zeigt die laufenden Prozesse
- -s zeigt die Signale
- -u zeigt die Besitzer der Prozesse
- -v vm Format
- -w breite Ausgabe
- -x zeigt Prozesse, die von keinem Terminal kontrolliert werden
- -S addiert die Prozessorzeit der Kindprozesse zu den Eltern
- -t.xx zeigt nur die Prozesse, die von Terminal xx kontrolliert werden

Übung zu Prozesse

- Starten sie xclock fünf Mal im Hintergrund.
- Listen Sie die Jobnummern
- Listen Sie alle laufenden Prozesse und bestimmen sie deren Vater
- Killen sie die Jobs auf verschiedene Art und Weise und beachten Sie dabei die Rückmeldung des kill-Kommandos (n: Jobnummer des ersten xclock-Jobs):
 - Job [n] mit kill JOBNUMMER
 - Job [n+1] mit kill -9 PID
 - Job [n+2] mit kill -3 JOBNUMMER
 - Job [n+3] mit kill -15 PID
- Holen Sie Job [n+4] in den Vordergrund und beenden sie ihn anschließend
- Starten Sie xclock im Vordergrund und schicken Sie es anschließend in den Hintergrund
- Starten Sie ein weiteres Mal 5 xclock-Prozesse im Hintergrund und killen Sie alle 5 mit einem einzigen Kommandoaufruf.

Editoren unter Linux

- UNIX-Standard-Editoren
 - vi
 - ed
 - emacs
 - xemacs
- Suse 9.2 + KDE
 - Kate

Entwicklung von C-Programmen

- Sourcecode-Erstellung mit Editor
 - vi, ed
 - emacs, xemacs
- Compilieren und Linken des Codes mit gcc (Gnu-C-Compiler)
 - gcc source.c
 - gcc source.c -o output
 - weitere Optionen von gcc siehe Manual Pages

HelloWorld.c

```
/* Beispielprogramm HelloWorld.c */  
#include <stdio.h>  
int main (void)  
{  
    printf("Hello World\n");  
    exit (0);  
}
```