

## Aufgabe 4:

### Yash (11 Punkte)

Entwerfen und programmieren Sie ein Programm `yash` (**y**et **a**nother **s**hell), das ähnlich wie eine primitive Shell unter UNIX Kommandos ausführt.

Ihr Programm soll als Promptsymbol "`yash>`" ausgeben, daraufhin eine Zeile von der Standardeingabe lesen, das darin angegebene Kommando in einem hierzu erzeugten Prozeß ausführen (hierbei natürlich Optionen und Argumente korrekt übergeben), auf das Terminieren der Kommandoausführung warten und den Exit-Status auf dem Standard-Fehlerkanal ausgeben. Anschließend soll ein neues Promptsymbol ausgegeben und das nächste Kommando entgegengenommen werden. (**fork(2)**, **exec(2)**, **wait(2)**)

Um die vom System unterstützte maximale Länge der Kommandozeile zu ermitteln, können Sie die Funktion **sysconf(3)** verwenden.

Das Shell-Programm soll terminieren, wenn es End-of-File (CTRL-D) vom Standardeingabekanal liest.

#### Beispiel:

```
[fau08h] [bin.i386]> ./yash
yash> ls -l
...
Exitstatus [ls -l] = 0
yash> ^D
[fau08h] [bin.i386]>
```

Testen Sie Ihr Programm auch mit fehlerhaften Eingaben, wie etwa ungültigen Programmnamen.

#### Zusatzaufgabe: POSIX-Signalmechanismus erklären (2 Zusatzpunkte)

Bereiten Sie sich auf die Tafelübungen der Woche vom 6.12.-10.12.99 auf die folgenden Fragestellungen vor:

- Wie kann man in einem Programm Vorbereitungen für das Eintreffen von Signalen treffen und welche Möglichkeiten der Reaktion auf Signale gibt es? (**sigaction(2)**)
- Wie kann man einem anderen Prozeß ein Signal zustellen? (**kill(2)**)
- Worin unterscheiden sich grundlegend die Signale SIGQUIT und SIGSEGV - wodurch werden die Signale ausgelöst?

**Abgabe: bis spätestens Donnerstag, 09.12.99, 14:00 Uhr**