

## Aufgabe 3:

### mini\_sh (12 Punkte) Bearbeitung in Zweier-Gruppen

#### a) Shell

Entwerfen und programmieren Sie ein Programm mini\_sh (**minimal shell**), das ähnlich wie eine primitive Shell funktioniert und Programme (im Weiteren als Kommandos bezeichnet) ausführt.

Ihr Programm soll als Promptsymbol das aktuelle Arbeitsverzeichnis (**getcwd(3)**) gefolgt von einem ":"-Zeichen ausgeben und anschließend eine Zeile von der Standardeingabe einlesen. Die eingelesene Zeile soll in Kommandoname und Argumente zerlegt werden, als Trennzeichen sollen Leerzeichen und Tabulatoren dienen (**strtok(3)**). Das Kommando soll dann in einem neu erzeugten Prozess (**fork(2)**) mit korrekt übergebenen Argumenten ausgeführt werden (**exec(2)**).

Ihr Programm soll anschließend auf das Terminieren der Kommandoausführung warten(**wait(2)**), und den Exitstatus ausgeben. Bei der Ausgabe soll unterschieden werden, ob der Prozess sich aktiv selbst beendet hat, oder ob der Prozess durch ein Signal beendet wurde und, wenn das Makro **WCOREDUMP** definiert ist, ob ein core-dump erzeugt wurde. Die Ausgabe soll wie folgt aussehen.

1. Fall: Prozess beendet sich selbst:

```
/proj/i4sos/sitesty: ls -l
...
Exitstatus [ls -l] = 0
```

2. Fall: Prozess wird durch ein Signal beendet:

```
/proj/i4sos/sitesty: ./sos1_wait
Signal [sos1_wait] = 2
```

3. Fall: Prozess wird durch ein Signal beendet und erzeugt einen core-dump:

```
/proj/i4sos/sitesty: ./sos1_wait
Signal [sos1_wait] = 3 (core dumped)
```

Nach der Ausgabe des Exitstatus soll die Shell wieder eine neue Eingabe entgegennehmen. Das Shell-Programm soll terminieren, wenn es beim Lesen vom Standardeingabekanal ein End-of-File (CTRL-D) erhält.

#### b) Verzeichniswechsel

Implementieren Sie nun noch einen Verzeichniswechsel (**chdir(2)**). Wird als Kommando "cd" eingegeben, so soll der mini\_sh-Prozess sein Arbeitsverzeichnis auf den im nachfolgenden Argument angegebenen Pfad setzen.

#### c) Makefile

Erstellen Sie ein zur Aufgabe passendes Makefile.

**Abgabe: bis spätestens Mittwoch, 24.05.2006, 17:30 Uhr**

Hinweise zur Lösung dieser Aufgabe:

- Im Verzeichnis */proj/i4sos/pub/aufgabe3* finden Sie die Programme *mini\_sh* und *sos1\_wait* zum Vergleichen bzw. Testen.
- Sie dürfen davon ausgehen, dass eine Kommandozeile maximal 1024 Zeichen lang ist. Beachten Sie jedoch, dass im Falle einer längeren Eingabe kein Speicherbereich ungewollt überschrieben wird.
- **WCOREDUMP** ist eine BSD-Erweiterung, welche beim Übersetzen mit -ansi standardmäßig nicht definiert ist. Übersetzen Sie ihr Programm darum zum Testen mit der zusätzlichen Option **-D\_BSD\_SOURCE**.