Prof. Dr. Franz Schweiggert Abteilung Angewandte Informationsverarbeitung
Ralph Guderlei

Systemnahe Software / Systemnahe Software II (SS 2005)

Abgabetermin: 07. Juni 2005

1 Pipes synchronisieren (10 Punkte)

Beim Arbeiten mit mehreren Pipes (oder später mit Sockets ...) kommt es vor, dass nicht jede Pipe sofort Daten liefert. Der system call `select` stellt einen Mechanismus zur Verfügung, um mehrere Filedeskriptoren zu beobachten.

Schreiben Sie ein Programm welches mehrere (<= 15) child - Prozesse erzeugt, die mittels Pipes mit dem parent-prozess kommunizieren (Sinnvollerweise verwendet jeder child-Prozess seine eigene Pipe). Doch statt sofort in die pipe zu schreiben, legt sich jeder child - Prozess für eine zufällige Zeitspanne (<=15s)schlafen. Pseudo-Zufallszahlen können Sie mit der Funktion `rand()` erzeugen. Davor initialsieren Sie den Zufallszahlengenerator am Besten mit `srand(time(NULL) + getpid())`, damit nicht jeder child - Prozess gleich lang schläft. Nach dem Aufwachen soll jeder child - prozess die aktuelle Zeit an den parent - prozess übermitteln.

Der parent - prozess soll mittels `select` diejenigen pipes ermitteln, aus denen gelesen werden kann. Wenn aus keiner pipe gelesen werden kann, soll der parent-prozess ebenfalls schlafen gelegt werden.

Tipp: Beachten Sie die Fehler die read() verursacht, wenn versucht wird aus einer leeren pipe zu lesen.

Interessante `man` pages:

- Linux: select (2), Solaris select (3c)
- select_tut (2), nur Linux oder Google ;-)
- read (2)
- rand / srand, Linux (3), Solaris (3c)
- time (2)
- ctime, Linux (2), Solaris (3c)

Viel Erfolg!