Universität Ulm - Abteilung Angewandte Informationsverarbeitung
Prof. Schweiggert und Dr. J. Mayer mit H. Braxmeier und C. Ehrhardt, Sommersemester 2004

6. Übungsblatt zur Vorlesung Systemnahe Software II

Abgabetermin: Montag, 14.06.2004

Poll und Pipes *(10 Punkte)*

Poll: Lesen von mehreren Filedeskriptoren

Häufig steht man (vorallem auch im Zusammenhang mit Netzwerkanwendungen) vor dem Problem, daß man nicht nur von einem Filedeskriptor Daten entgegen nehmen möchte, sondern von mehreren. Allerdings kann ein normales Lesen von einem der Filedeskriptoren dazu führen, daß der Prozeß blockiert wird, wenn dort gerade keine Daten zur Verfügung stehen. Solange der Prozeß blockiert ist, können dann auch Daten, die auf anderen Filedeskriptoren zur Verfügung stehen nicht bearbeitet werden.
Es wäre also praktisch, wenn man vor dem eigentlichen Lesen feststellen könnte, ob von einem Filedeskriptor gelesen werden kann, ohne daß der Prozeß dadurch blockiert wird. Analog gilt das natürlich auch beim Schreiben.

Der `poll`-Systemaufruf

Der Systemaufruf `poll` (siehe auch "man poll") ist hierbei hilfreich. Diesem Systemaufruf wird eine Liste von Filedeskriptoren übergeben. Für jeden Filedeskriptor wird zusätzlich angeben, was wir mit diesem Filedeskriptor machen wollen (lesen, schreiben oder beides). Der Systemaufruf `poll` blockiert dann den Prozeß so lange, bis mindestens eine der Aktionen ohne Blockieren durchgeführt werden kann. Zusätzlich kann noch ein Zeitlimit angegeben werden, dann wartet `poll` höchstens so lange. (Wie bei blockierenden Systemaufrufen üblich wird `poll` zusätzlich auch von einem eintreffenden Signal unterbrochen.) Der Rückgabewert gibt an, wieviele Filedeskriptoren die gewünschte Operation jetzt erlauben. Um welche Filedeskriptoren es sich dabei genau handelt kann dann der übergebenen Liste entnommen werden (siehe nächster Abschnitt).

Die pollfd-Datenstruktur

Jeder Filedeskriptor in der oben erwähnten Liste wird durch eine Struktur vom Typ `struct pollfd` beschrieben. Diese Datenstruktur enthält drei Felder:

fd Die Nummer des Filedeskriptors. Wenn dieser Wert kleiner als 0 ist, wird dieser Listeneintrag von `poll` ignoriert.

events Hier wird durch verschiedene Flags angegeben, welche Aktion wir gerne mit dem Filedeskriptor durchführen würden. Interessant für uns sind die Flags:

POLLIN Wir würden gerne lesen

POLLOUT Wir würden gerne schreiben

revents Dieses Feld wird von `poll` verwendet, um uns nach der Rückkehr des Systemaufrufs mitzuteilen, welche der verlangten Aktionen jetzt mit diesem Filedeskriptor ohne Blockieren durchgeführt werden können. Wenn POLLIN gesetzt ist, dann kann mindestens ein Zeichen gelesen werden, wenn POLLOUT gesetzt ist, dann kann mindestens ein Zeichen geschrieben werden. Zusätzlich kann noch das Flag POLLHUP gesetzt sein, wenn das andere Ende der Dateiverbindung (z.B. bei einer Pipe) geschlossen wurde.

Aufgabe: Reaktionszeit beim Lesen

Kindprozeß

Zunächst soll eine Prozedur geschrieben werden, die einen Kindprozeß erzeugt, der über eine pipe mit dem Vater verbunden ist. Dieser Kindprozeß soll in einer Schleife eine zufällige Zeit zwischen 2 und 10 Sekunden schlafen, dann einen ebenfalls zufälligen Buchstaben in die Pipe zum Vater schreiben. Die Schleife soll zwischen 3 und 5 mal durchlaufen werden (ebenfalls zufällig ausgewählt). Jedes Mal, wenn der Kindprozeß ein Zeichen in die Pipe schreibt, soll er dieses Zeichen auch zusammen mit seiner Prozeß-ID auf die Standardausgabe ausgeben.
Hinweis: Die Funktion `rand` erzeugt Zufallszahlen. Es empfiehlt sich in **jedem** neu erzeugten Prozeß vor dem ersten Aufruf von `rand` einmal `srand(time(0) + getpid());` aufzurufen.

Vaterprozeß

Der Vaterprozeß soll gleichzeitig 5 Kindprozesse wie oben beschrieben starten, von denen jeder über eine eigene Pipe mit dem Vater kommuniziert. Sobald von einer dieser Pipes ein Zeichen gelesen werden kann, soll der Vater dieses Zeichen lesen und zusammen mit der Prozeß-ID des Kindprozeßes, von dem er es gelesen hat ebenfalls auf die Standardausgabe ausgeben. Ziel ist es, daß die Ausgabe des Vaters und die zugehörige Ausgabe des Kindprozesses zeitgleich erfolgen! Dazu soll `poll` wie oben beschrieben verwendet werden.

Viel Erfolg!