Dr. Matthias Grabert
Christian Ehrhardt Abteilung Angewandte Informationsverarbeitung 6. Mai 2002
Blatt 3

Systemnahe Software (SS 2002)

Abgabetermin 14.05.2002

Synchronisation

In diesem Blatt soll ein Dämon zur Verwaltung einer bestimmten Resource geschrieben werden. Der Dämon soll sicherstellen, daß immer nur ein Prozeß auf diese Resource zugreift. Er ist dazu auf die Zusammenarbeit der zugreifenden Prozesse angewiesen.

Der Ablauf

Für die Synchronisation werden drei Signale benötigt:

REQUEST Dieses Signal wird von einem Prozeß, der auf die Resource zugreifen will, an den Dämon geschickt. Der Dämon merkt sich (z.B. in einer FIFO-Queue), von welchen Prozessen er solche Anforderungen bekommen hat. Sobald die Resource frei wird, teilt er dem ersten wartenden Prozeß aus der FIFO die Resource zu.

GRANT Dieses Signal wird vom Dämonprozeß verschickt, um einem wartenden Prozeß mitzuteilen, daß die Resource jetzt für ihn reserviert ist.

DONE Dieses Signal wird von dem Prozeß, dem die Resource gerade zugeteilt ist, an den Dämon verschickt, wenn er die Resource nicht mehr benötigt. Der Dämon sollte dieses Signal nur von dem Prozeß akzeptieren, dem er gerade die Resource zugeteilt hat.

Siginfo

Traditionell war es für einen Prozeß nicht möglich zu erfahren, von welchem Prozeß ein Signal veranlaßt wurde. Seit einigen Jahren gibt es aber eine Möglichkeit, mehr Informationen über den Ursprung eines Signals zu bekommen. Dazu muß bei sigaction das Flag SA_SIGINFO mit angegeben werden. Der Signalhandler bekommt dann als zweiten Parameter (der erste ist immernoch die Signalnummer) einen Zeiger auf eine Struktur (struct siginfo *), die weitere Information enthält. Die Details dazu sollten den Manualseiten für sigaction (2) und siginfo (3HEAD) entnommen werden.

Signalnummern

In signal.h wird die Konstante SIGRTMIN definiert. Für die drei oben beschrieben Signale sollten die Nummern SIGRTMIN, SIGRTMIN+1 und SIGRTMIN+2 verwendet werden.

Blockieren von Signalen

Signale können zu jedem Zeitpunkt auftreten. Wenn das zu unerwünschten Effekten führen kann, dann kann die Funktion sigprocmask(2) verwendet werden um Signale vorübergehend zu blockieren. Innerhalb eines Signalhandlers kann das Blockieren von Signalen statt dessen mit Hilfe von sa_mask (siehe man sigaction) erreicht werden.

Ein Client

Damit die Synchronisation getestet werden kann soll auch ein Client geschrieben werden, der

- die Resource anfordert,
- auf die Zuteilung wartet,
- 5 Sekunden schläft und
- die Resource wieder freigibt.

Die Prozeß-ID das Dämonprozesses kann z.B. auf der Kommandozeile übergeben werden. Der Client sollte bei den einzelnen Schritten kurze Statusmeldungen ausgeben. Mit einigen (beinahe) gleichzeitig gestarteten Clients kann dann getestet werden, ob die Synchronisation auch tatsächlich funktioniert.

Nützliche Manualseiten

Der Umgang mit dem Typ sigset_t, also mit einer Menge von Signalen, wird in sigemptyset (3C) erklärt. Unter Linux wird struct siginfo auf der Manualseite von sigaction mit erklärt.