

4. Übung

Ausgabe Abgabe Diskussion
08.11.13 15.11.13 18.11.-22.11.13

Bitte bei der Abgabe Name/Matr.Nr. der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind.

Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Prozesszustände (4+1+1=6 Punkte)

In der Vorlesung haben Sie gelernt, dass sich Prozesse während ihrer Lebensdauer in verschiedenen Zuständen befinden können:

Zustand	Bedeutung
new	Prozess wird erzeugt (kreiert)
running	Anweisungen des Prozesses werden gerade ausgeführt
ready	Prozess ist bereit zur Ausführung, wartet aber noch auf freien Prozessor
waiting	Prozess wartet auf das Eintreten eines Ereignisses, z.B. darauf, dass ein von ihm belegtes Betriebsmittel fertig wird
blocked	Prozess wartet auf ein fremdbelegtes Betriebsmittel
killed	Prozess wird (vorzeitig) abgebrochen
terminated	Prozess ist beendet, d.h. alle Instruktionen sind abgearbeitet

Gegeben seien 4 Prozesse, die zu unterschiedlichen Zeitpunkten gestartet werden und sich in die ready-Warteschlange einreihen. Der Scheduler arbeitet nach dem FIFO-Verfahren, d.h. der Prozess, der zuerst auf **ready** steht, wird zuerst bedient.

Einige Prozesse benötigen Zugriff auf den Drucker. Die zweite Zeile der Tabelle gibt an, nach wie vielen Zeiteinheiten im Zustand **running** ein Prozess auf den Drucker zugreifen möchte. Während der 5 Zeiteinheiten dauernden Druckphase wird der Prozessor für andere Prozesse freigegeben. Nach Beendigung des Druckvorgangs reiht sich der Prozess wieder in die ready-Warteschlange ein. Der Prozess P2 wird abgebrochen, nachdem er 2 Zeiteinheiten gerechnet hat.

Prozess	P1	P2	P3	P4
Startzeitpunkt	0	1	2	3
Drucken nach Zeiteinheit	4	6	1	-
Benötigte Zeiteinheiten im Zustand running	8	10	4	5

- Geben Sie zu den Zeitpunkten 0, 1, 2, ... den aktuellen Prozesszustand der Prozesse P1 bis P4 an. Beachten Sie, dass ein ankommender Prozess bei seiner Ankunft den Zustand **new** annimmt und erst nach einer Zeiteinheit in einen anderen Zustand wechselt, wobei keine CPU-Zeit verbraucht wird.
- Zu welchen Zeitpunkten befindet sich das System im Zustand des *busy waiting*, d.h. ein Prozess wartet auf ein Betriebsmittel und blockiert derweil die CPU?

- c) Nennen Sie ein Beispiel, bei dem der Abbruch eines Prozesses sinnvoll sein kann. Wie kann man unter dem Betriebssystem UNIX Prozesse abbrechen?

Aufgabe 2: Fragmentierung (1+1=2 Punkte)

Erklären Sie die Auswirkungen von externer Fragmentierung auf den Hauptspeicher und auf Festplatten.

Aufgabe 3: Speicherzuteilung (1+1=2 Punkte)

Wie oft können Sie unter Linux `malloc(1000000000000)` aufrufen, wenn ihr System 1 GB RAM und ein 2 GB Swapfile hat. Recherchieren und Begründen Sie!

Aufgabe 4: Einführung in C: Filekonzept (1+2+2=5 Punkte)

Diese Aufgabe soll Sie mit dem Filekonzept von C bekannt machen.

Es gibt verschiedene C-Funktionen, mit denen Sie auf Dateien zugreifen können. Zwei häufig verwendete Funktionen sind `fopen()` zum Öffnen und `fclose()` zum Schließen einer Datei. Nachdem eine Datei erfolgreich geöffnet worden ist, können Sie Daten aus der Datei lesen bzw. in die Datei schreiben. Das von `fopen()` gelieferte Filehandle ist solange gültig, bis Sie die Datei wieder schließen.

- Studieren Sie zunächst den Quelltext von `u4.4.c`. Was leistet das Programm? Kommentieren Sie im Quelltext die wichtigsten Abschnitte.
- Welche 3 Grundtypen gibt es, eine Datei mit `fopen()` zu öffnen? Worin unterscheiden sich diese Typen? Modifizieren Sie den Quelltext von `u4.4.c` derart, dass die Ausgabedatei (so vorhanden) nicht überschrieben wird, sondern dass die Daten an das Dateiende angehängt werden.
- Was passiert, wenn eine Datei mit dem Zugriffstyp `rb` anstatt `rt` geöffnet wird? Ändern Sie den Zugriffstyp beim Öffnen der Eingabedatei in `u4.4.c`. Übersetzen Sie das Programm noch einmal und prüfen Sie, ob sich die Funktionalität des Programms geändert hat. Begründen Sie ihre Beobachtung.