

Übungsblatt 3: Scheduling

⇒ Abgabe der Lösungen bis Montag, 08. November 14:00 im AsSESS

Aufgabe 1: Zombies, C und Nettigkeiten (1+1+1 = 3 Punkte)

- Beschreiben Sie im Kontext von UNIX-Prozessen den Unterschied zwischen Zombies und verwaisten Prozessen.
- Erklären Sie den Unterschied der Wirkung des Schlüsselworts *static* auf globale bzw. lokale Variablen in einem C-Programm.
- Erklären Sie, welchen Effekt *nice* im folgenden Shell-Aufruf auf die Ausführung von *grep* hat:

```
nice -n 15 grep -r needle ~
```

Welche Gründe könnte es für den Einsatz von *nice* in diesem Aufruf geben?

→ antworten.txt

Aufgabe 2: SPN (4+1 = 5 Punkte)

Betrachten Sie die folgenden drei Prozesse, die gleichzeitig in die Ready-Liste aufgenommen werden (Ankunftszeit 0). Zur Vereinfachung sei angenommen, dass die CPU- und E/A-Stöße pro Prozess immer gleich lang sind. Die Prozesse führen jeweils periodisch erst einen CPU-Stoß und dann einen E/A-Stoß der angegebenen Dauer durch (also CPU → E/A → CPU → E/A → ...). Alle Zeiten sind in Millisekunden angegeben.

Prozess-ID	CPU-Stoßlänge	E/A-Stoßlänge
A	5	1
B	1	1
C	2	1

Sie können AnimOS (<https://ess.cs.uos.de/software/AnimOS/> → CPU-Scheduling) verwenden, um sich einen ersten Überblick über die verschiedenen Schedulingverfahren zu verschaffen.

Achtung: In der Klausur steht Ihnen AnimOS nicht zur Verfügung. Sie sollten die Lösungen daher auch ohne AnimOS erarbeiten können.

- Wenden Sie auf die Prozesse A, B und C das *Shortest-Process-Next*-Scheduling-Verfahren aus der Vorlesung an. Gehen Sie dabei davon aus, dass dem Scheduler die Laufzeiten der einzelnen Prozesse bekannt sind. Notieren Sie die CPU- und E/A-Verteilung für die ersten 30 ms. (Annahmen: Prozesswechsel sind zu vernachlässigen; es können mehrere E/A-Vorgänge parallel ausgeführt werden.) Stellen Sie Ihre Ergebnisse wie in der folgenden ASCII-Zeichnung dar. Eine Spalte entspricht dabei 1 ms. Nutzen Sie dazu in Ihrem Editor eine Monospace-Schriftart und keine Tabulatoren.

```
+---+-----+
| A |CCCCC-----| C: Prozess nutzt die CPU
| B |-----CCEEE ... | E: Prozess führt E/A-Operationen durch
| C |-----CCC    | -: Prozess ist lafbereit
+---+-----+
```

2. Welches Problem birgt *SPN* als Scheduling-Verfahren?

→ antworten.txt

Aufgabe 3: Round-Robin (2+1 = 3 Punkte)

Unter den gleichen Voraussetzungen wie in Aufgabe 3 soll nun das *Round-Robin*-Scheduling-Verfahren auf die folgenden Prozesse angewandt und der Einfluss der Zeitscheibenlänge auf den Schedule untersucht werden.

Prozess-ID	CPU-Stoßlänge	E/A-Stoßlänge
A	19	1
B	2	8
C	5	5
D	2	2

Simulieren Sie die angegebene Prozessmenge in AnimOS für Zeitscheiben der Länge 10ms, 5ms, 2ms und 1ms. Lassen Sie jeweils unter dem Punkt „Statistics“ über die Schaltfläche „Create statistics for selected schedule“ den erzeugten Schedule auswerten.

1. Geben Sie für jeden Prozess die Zeitscheibenlänge an, für die dem Prozess der größte CPU-Anteil zugewiesen wird. Wie lässt sich das Ergebnis anhand der jeweiligen Prozess-Eigenschaften erklären?
2. Bei einer Zeitscheibenlänge von 1ms weist die Statistik die geringste mittlere Antwortzeit der untersuchten Schedules aus. Was könnte in der Praxis dagegen sprechen, eine besonders kleine Zeitscheibenlänge zu wählen?

→ antworten.txt