

Prüfungsklausur Betriebssysteme und Sicherheit am 27. Februar 2012

Alle Aussagen sind so ausführlich wie nötig, aber so knapp wie möglich zu begründen.

Bewertung 1: 2: 3: 4: 5: 6: Σ :

1. Aufgabe

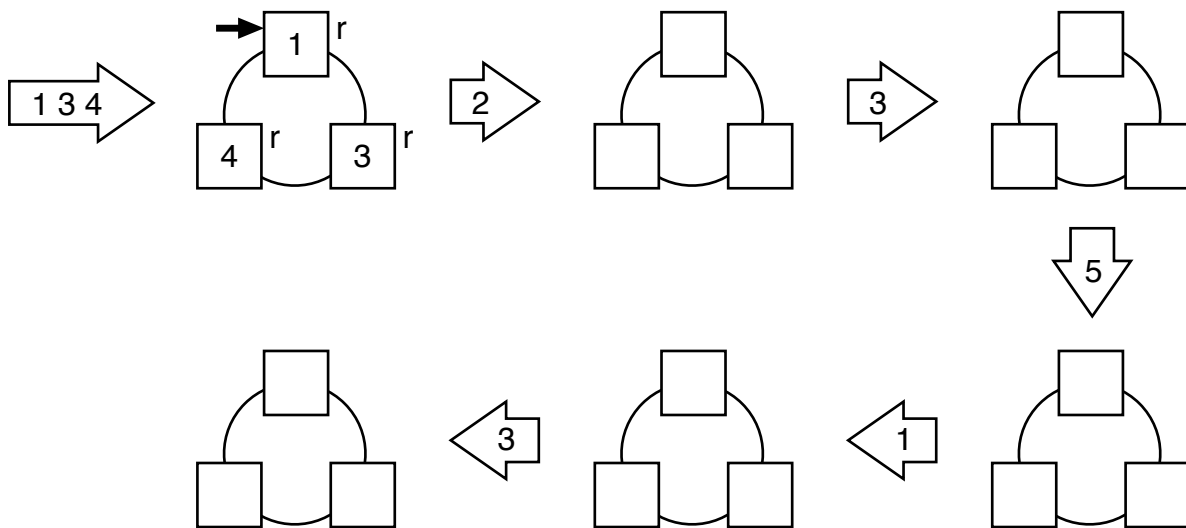
7 Punkte

Ein System hat 3 Kacheln. Geben Sie das Verhalten der Verdrängungsstrategien *Clock* und *Least Recently Used (LRU)* für folgende Sequenz von Seitenzugriffen an:

1 3 4 2 3 5 1 3

Welche Vor- und Nachteile hat *LRU* im Vergleich zu *Clock*?

Clock:



LRU:

Kachel	1	3	4	2	3	5	1	3
1	1							
2		3						
3			4					

2. Aufgabe**5 Punkte**

Betrachtet wird ein System, dessen virtueller und physischer Adressraum jeweils mit 32 Bit breiten Adressen angesprochen wird. Die Seitengröße beträgt 4 KiB, die Seitentabelle ist zweistufig. Die einzelnen Tabellen benötigen jeweils 4 KiB Speicherplatz und enthalten jeweils 1024 Einträge.

Hinweis: Die Einheit KiB bezeichnet 1024 Byte.

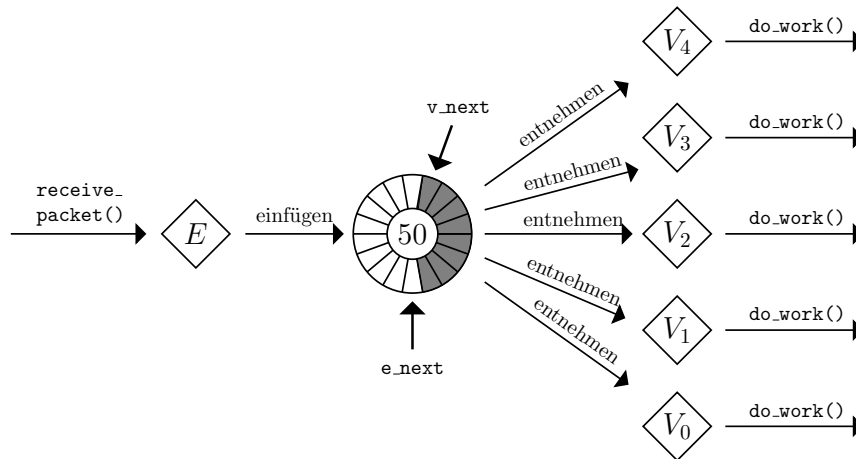
- a) Wie viele Einträge werden in beiden Stufen benötigt um den gesamten virtuellen Adressraum abzubilden?
- b) Eine Anwendung benötigt ein 11 KiB großes Code-Segment und ein 25 KiB großes Datensegment. Das Datensegment soll schreibbar, das Code-Segment nicht schreibbar sein. Das Betriebssystem verwaltet Speicherbelegungen nur in ganzen Seiten, ansonsten dürfen die Segmente beliebig im virtuellen Adressraum platziert werden.

Wie viel physischer Speicher wird im günstigsten und im ungünstigsten Fall benötigt? Berücksichtigen Sie sowohl den Speicherbedarf der Anwendung selbst als auch den Speicher, der für die Seitentabellen erforderlich ist.

3. Aufgabe**9 Punkte**

Ein Webserver bestehe aus mehreren zusammen arbeitenden Threads. Ein Eingabe-Thread E empfängt Anfragen aus dem Netzwerk mittels einer Funktion `receive_packet()` und fügt diese an Position `e_next` in einen Ringpuffer ein. Eine Gruppe von Verarbeitungs-Threads V_i entnimmt Anfragen an Position `v_next` aus dem Ringpuffer und verarbeitet sie mittels einer Funktion `do_work()`. Dabei wird jede Anfrage von genau einem Thread bearbeitet.

Der Ringpuffer hat eine Kapazität von 50 Anfragen. Es laufen fünf Verarbeitungs-Threads V_0 bis V_4 .



Geben Sie in Pseudo-Code eine Implementierung der Thread-Funktionen für E und V_i an, welche

- die Ringpuffer-Positionen `e_next` und `v_next` geeignet verwaltet,
- größtmögliche Parallelität zwischen allen Threads gewährleistet,
- ohne Busy Waiting auskommt und
- gängige Probleme bei der parallelen Verarbeitung (Wettlaufsituationen, Verklemmungen) meidet.

4. Aufgabe**7 Punkte**

In einem Echtzeitsystem ist eine Menge von drei periodischen Tasks so einzuplanen, dass deren Jobs in jeder Periode erfolgreich beendet werden. Das Periodenende entspricht der Deadline. Die Parameter der Tasks beschreiben jeweils die Periode p und die konstante Bearbeitungszeit e der Jobs:

$$T_1: p_1 = 12, e_1 = 4; \quad T_2: p_2 = 6, e_2 = 2; \quad T_3: p_3 = 4, e_3 = 1$$

Die Tasks sind unabhängig voneinander und an beliebiger Stelle unterbrechbar. Der Scheduling-Overhead werde vernachlässigt. Es stehe genau ein Prozessor zur Verfügung.

- Gibt es ein Scheduling-Verfahren, welches die gegebene Taskmenge unter den genannten Bedingungen einplanen kann?
- Ist die gegebene Taskmenge mit statischen Prioritäten einplanbar?
- Während des Entwurfs des Echtzeitsystems wird festgestellt, dass die Periode der Task T_3 verändert werden muss. Welche Periode p_3 kann minimal verwendet werden, so dass die Taskmenge mit dynamischen Prioritäten einplanbar ist?
- EDF ist ein Scheduling-Verfahren, welches dynamische Prioritäten verwendet. Für die Implementierung eines Schedulers in einem eingebetteten System entscheidet der Systemarchitekt, dass dynamische Prioritäten zu aufwändig sind. Er schlägt folgende Modifikation vor:

Zum Zeitpunkt 0 des Systems werden alle Tasks gleichzeitig bereit. Zu diesem Zeitpunkt werden einmalig Prioritäten nach EDF an die Tasks vergeben. Anschließend sind diese Prioritäten statisch, werden also nicht mehr verändert.

Geben Sie diese Prioritätszuteilung für obige Taskmenge an. Vergleichen Sie das vorgeschlagene Verfahren bezüglich Einplanbarkeit mit EDF.

Sie können diesen Bereich zum Skizzieren von Ablaufplänen nutzen:



5. Aufgabe**7 Punkte**

Alice hat den folgenden öffentlichen RSA-Schlüssel, dabei bezeichnet n_A den Modul und c_A den Exponenten:

$$n_A = 65, \quad c_A = 29$$

- a) Verschlüsseln Sie die Botschaft $x = 63$.
b) Bob hat den folgenden öffentlichen Schlüssel:

$$n_B = 91, \quad c_B = 29$$

Bestimmen Sie $q = \text{ggT}(n_A, n_B)$. Welches Sicherheitsproblem deckt das Ergebnis auf?

- c) Bestimmen Sie den privaten Schlüssel von Alice.

6. Aufgabe**7 Punkte**

Gegeben ist folgende C-Funktion `f`, die auf einem UNIX-System ausgeführt werden soll:

```
void f(int fd, unsigned char c) {
    if (c > 0) {
        pid_t pid = fork();
        if (pid > 0)
            wait(0);
        write(fd, &c, 1);
        f(fd, c-1);
    }
}
```

Der Aufruf der Funktion erfolgt durch `f(fd, 2)`. Dabei verweist der im ersten Argument übergebene File Descriptor `fd` auf eine bereits geöffnete, leere Datei.

Hinweise: Der Aufruf `wait(0)` wartet blockierend auf die Beendigung eines beliebigen Kindprozesses.

Gehen Sie außerdem davon aus, dass alle Systemaufrufe keinen Fehlerwert zurück liefern.

- a) Skizzieren Sie grafisch den zeitlichen Ablauf der beteiligten Prozesse.
- b) Geben Sie eine mögliche Folge von Werten an, die nach Abschluss aller durch `f` ausgelösten Schreibzugriffe in der mit `fd` verbundenen Datei stehen kann.