

Übungsblatt 4 - Betriebssysteme

Aufgabe 1 (Peterson-Algorithmus)

Beweisen Sie, dass der Peterson-Algorithmus verklemmungsfrei ist.

Hinweis: Im Skript wird bewiesen, dass der Algorithmus gegenseitigen Ausschluss garantiert. Der Beweis der Verklemmungsfreiheit kann in ähnlicher Weise geführt werden.

```
#define FALSE 0
#define TRUE 1
#define N 2 /* number of processes */

int turn; /* whose turn is it ? */
int interested[N] /* all values initially 0 (FALSE) */

void enter_region(int process) /* process: who is entering (0 or 1) */
{
    int other; /* number of the other process */

    other = 1-process;
    interested[process] = TRUE /* show that you are interested */
    turn = process; /* set flag */
    while ( turn == process && interested[other] == TRUE )
        ; /* busy waiting */
}

void leave_region(int process) /* process: who is leaving (0 or 1) */
{
    interested[process] = FALSE;
    /* indicate departure from critical region*/
}
```

Aufgabe 2 (Speicherverwaltung)

Betrachten Sie das nachfolgende C-Programm

```
#define SPALTEN 8192
#define ZEILEN 100000
int feld[ZEILEN][SPALTEN];
int main(){
    register int i,j;
    for ( i=0; i<ZEILEN; i++ )
        for ( j=0; j<SPALTEN; j++ )
            feld[i][j] = 1;
}
```

a) Wieviel Speicherplatz wird das Programm etwa benötigen?

- b) Im logischen Adressraum des Prozesses ist Speicherplatz für „feld“ verfügbar. Zu welchem Zeitpunkt wird dieser Speicherplatz bereitgestellt?
- c) Halten Sie das Programm irgendwie vor und nach der Schleife an und überprüfen Sie in der Prozesstabelle jeweils den realen Speicherbedarf. Stimmen die Werte mit ihrer Erwartung überein. Erklären Sie die Werte.
- d) Messen Sie die Zeit für die Feld-Initialisierung mit dem Programm *time*. Von der angezeigten „user“-Zeit wird ein Teil auf die CPU-internen Berechnungen und ein Teil auf die Hauptspeicherzugriffe entfallen. Wie könnte man den für die Berechnungen benötigte Zeit ermitteln? Wie lange dauert ein Hauptspeicherzugriff im Durchschnitt? Wieviele Systemtakte sind das?
- e) Erwarten Sie, dass sich die Laufzeit ändert, wenn man das Feld spaltenweise bearbeitet?

```
for ( j=0; j<SPALTEN; j++ )
  for ( i=0; i<ZEILEN; i++ )
    feld[i][j] = 1;
```

Begründen Sie ihre Antwort und überprüfen Sie es durch eine Zeitmessung.

- f) Bei einem modernen System wird sich die Laufzeit bei spaltenweiser Bearbeitung des Felds nicht nur verdoppeln sondern vielleicht verzehnfachen. Dies ist nicht durch den TLB-Verzicht erklärbar. Welche Effekte spielen hier noch mit?
- g) In welcher Situation konnte sich die Laufzeit durch die spaltenweise Bearbeitung ver Hundertfachen?

Aufgabe 3 (Virtueller Speicher)

Ein System verwendet virtuellen Speicher mit Paging. Für die Hauptspeicherverwaltung wird eine dreistufige Seitentabelle benutzt. Physikalische Adressen sind 64-Bit groß. Eine virtuelle Adresse ist 32 Bit groß und von der Form

| | | | |
|-------|-------|-------|-----|
| p_1 | p_2 | p_3 | D |
|-------|-------|-------|-----|

mit folgender Aufteilung:

- p_1 : 5 Bit für die Seitentabelle der 1. Stufe
- p_2 : 8 Bit für die Seitentabelle der 2. Stufe
- p_3 : 8 Bit für die Seitentabelle der 3. Stufe
- D : 11 Bit für die Distanz

Annahme: Der virtuellen Adresse $v=0x00000901$ entspricht die reale Adresse $r=0x001010000000101$

- a) Wie groß sind die Seitenrahmen für das obige Beispiel?
- b) Geben Sie an, wie der TLB-Eintrag zu (v,r) aussieht
- c) Geben Sie an, wie die Rahmennummer im Beispiel bestimmt wird. Zeichnen Sie dazu eine Skizze mit den benötigten Seitentableneinträgen.
- d) Die Umrechnung von virtuellen in reale Speicheradressen in einem System mit Paging kostet Zeit.
Dauert die Berechnung einer realen Adresse in einem Rechner mit 3-stufigen Seitentabellen **im Durchschnitt wesentlich** länger als in einem Rechner mit 2-stufigen Seitentabellen (Begründung)?