

Vorlesung Betriebssystemarchitektur WS 2002/03

Aufgabenblatt 5 vom 18. Dezember 2002

(Vorstellung der Lösungen bei den Tutoren bis zum 17. Januar 2003)

Aufgabe 5.1: (25 Punkte)

In dieser Aufgabe sollen Sie das Zeitverhalten des Betriebssystems untersuchen. Genauer: Sie sollen die Zeit ermitteln, die das Betriebssystem für den Wechsel von einem Prozess zum nächsten benötigt. Kompilieren Sie dazu die Programme **event1** und **event2** unter Windows 2000 (Quelldateien sind auf der Webseite <http://www.dcl.hpi.uni-potsdam.de/uebung2002/blatt5> zu finden). Das Programm **event1** öffnet zwei benannte Event-Objekte "PingEvent" und "PongEvent", auf die das Programm **event2** ebenfalls zugreift. Das Programm **event1** löst "PingEvent" aus und wartet dann auf ein "PongEvent". Das Programm **event2** verhält sich genau entgegengesetzt, sodass ein ständiges hin und her erzwungen wird. Beide Programme geben Werte des Timestamp Counter Registers des Prozessors aus. Der Timestamp Counter wird mit jedem Prozessor-Taktzyklus um 1 inkrementiert und lässt sich deshalb als Uhr verwenden. Im Beispiel-Programm **auswertung** wird gezeigt, wie Sie zwei Dateien einlesen und auswerten können.

Um folgende Fragen beantworten zu können, sollen Sie ein eigenes Auswerte-Programm schreiben. Benutzen Sie zum Erzeugen der Messdaten die vorgegebenen Programme **event1** und **event2**. Führen Sie Ihre Messungen zunächst so durch, dass **event1** und **event2** die einzigen "richtig aktiven" Prozesse sind und nicht durch andere gestört werden.

- Wie lange dauert der Aufruf von `rdtsc()` durchschnittlich auf dem von Ihnen benutzten Rechner? (Angabe in Taktzyklen und in Sekunden - bitte Prozessor-Typ und Taktrate angeben)
- Wieviele Prozessor-Taktzyklen benötigt ein Umschaltvorgang von einem Prozess zum nächsten mindestens? Wie ermitteln Sie diese Umschalt-Zeit?
- Wie lange dauert der Umschaltvorgang im Durchschnitt? Geben Sie die Standardabweichung dieses Mittelwertes an.

Stellen Sie Ihre Messwerte geeignet graphisch dar (Histogramm, ggf. logarithmische Darstellung).

- Identifizieren Sie den Peak mit der höchsten Amplitude im Histogramm. An welcher Position befindet sich der Peak?
- Bestimmen Sie die Breite dieses Peaks (full width, half maximum: FWHM).
- Wieviel Prozent der Messwerte liegen in diesem Peak?
- Zeichnen Sie in Ihr Diagramm den oben berechneten Mittelwert ein und erklären Sie Ihrem Tutor, warum der Mittelwert über oder unter der Peak-Position liegt.

Wiederholen Sie die Messungen unter anderen Lastbedingungen des Rechners. Es sollen neben **event1** und **event2** noch drei weitere Prozesse in einer Endlosschleife laufen und damit maximal viel CPU-Zeit beanspruchen. Beantworten Sie die oben gestellten Fragen auch für diese Messung. Bringen Sie beide Diagramme zum Tutoriumstermin in Papierform mit und geben Sie sie bei Ihrem Tutor ab. Bitte Beschriften Sie die Diagramme aussagekräftig, d.h., inklusive Prozessor-Typ, Taktrate usw.

Aufgabe 5.2: (25 Punkte)

Wie Aufgabe 5.1, allerdings sollen die beiden Programme **event1** und **event2** nun nicht in zwei verschiedenen Prozessen laufen, sondern in einem Prozess in zwei Threads. Erklären Sie die Unterschiede zu den Ergebnissen der Aufgabe 5.1.

Aufgabe 5.3: (30 Punkte)

Erzeugen Sie eine Datei mit 7000 Bytes unter Windows 2000. Schreiben Sie ein C-Programm, das unter Verwendung von Win32-API-Aufrufen an die Byte-Position 4000 in dieser Datei einhundert Bytes im Sekundentakt schreibt. Die geschriebenen Daten sollen bei jeder Schreiboperation unterschiedlich sein, sodass sie einer bestimmten Schreiboperation zugeordnet werden können (z.B. 100 mal 0x00 beim ersten Schreiben, 100 mal 0x01 beim zweiten Schreiben usw.). Hinweis: lesen Sie sich die Beschreibung zu folgenden Funktionen in der MSDN-Hilfe durch: CreateFile, WriteFile, ReadFile, SetFilePointer, Sleep.

Schreiben Sie ein zweites C-Programm, das hundert Bytes ab Position 4000 aus dieser Datei alle 400ms ausliest. Lassen Sie beide Programme gleichzeitig laufen und beantworten Sie folgende Fragen:

- Wird jede Schreiboperation so ausgeführt, dass der Lese-Prozess alle Änderungen in der Datei mitbekommt?
- Legen Sie die Datei in drei unterschiedliche Dateisysteme: lokale Platte (NTFS), Diskette (FAT) oder Netzlaufwerk (CIFS). Gibt es Unterschiede?
- Legen Sie die Datei in ein Netzlaufwerk und lassen Sie Schreib- und Lese-Prozess auf unterschiedlichen Rechnern laufen. Bekommt der Lese-Prozess alle Datei-Änderungen mit?

Aufgabe 5.4: (20 Punkte)

Wie Aufgabe 5.3, allerdings diesmal unter Linux. Lesen Sie die Manuseiten zu open(2), read(2), write(2), lseek(2). Legen Sie die Datei im lokalen Dateisystem ab (z.B. in /tmp (EXT2-Filesystem)). Wiederholen Sie den Versuch in Ihrem Home-Verzeichnis (NFS).