Vorlesung Betriebssystemarchitektur WS 2007/08
Aufgabenblatt 4 vom 10. Januar 2008
(Vorstellung der Lösungen bei den Tutoren bis zum 23.01.2008)

Aufgabe 4.1: (10 Punkte)
Erläutern Sie Ihrem Tutor den Unterschied zwischen preemptiven und nicht-preemptiven Scheduling. Welcher Ansatz ist zu bevorzugen, wenn der Systemdurchsatz maximiert werden soll? Wie versuchen Windows-Serversysteme den Durchsatz zu maximieren?

Aufgabe 4.2: (30 Punkte)
Gegeben sei ein Einprozessor-System welches Round-Robin-Scheduling mit 16 Prioritätsstufen verwendet (0-15, 0 = niedrigste, 15 = höchste Priorität). Die Quantumslänge beträgt 20ms. Die Zeit für einen Kontextwechsel ist vernachlässigbar. Der Scheduler verwaltet laufende Threads und entscheidet ausschließlich nach dem Ablauf eines Quantums welcher Thread als nächstes laufen soll (d.h. es gibt keine Unterbrechungen).

Es sollen drei Threads mit den folgenden Eigenschaften ausgeführt werden:

Thread ID	Startzeit	Ausführungszeit
Th1	t = 0ms	100ms
Th2	t = 15ms	90ms
Th3	t = 30ms	70ms

- Zeichnen Sie ein Gantt-Diagramm unter der Annahme, dass alle drei Threads mit einer (statischen) Priorität von 8 ausgeführt werden.
- Jetzt soll Th3 mit einer Priorität von 9 ausgeführt werden. Weiterhin erhält Th1 eine Priorität von 7 und tritt nach 16ms in einen I/O-Wartezustand. Die Priorität von Th1 soll nach Beendigung der I/O-Operation um drei erhöht werden („*Boost*“). Th1 verlässt den Wartezustand bei t=45ms. Die Priorität von Th1 wird um eine Stufe am Quantumsende reduziert, bis wieder die Basispriorität erreicht wird.
  Zeichnen Sie ein Gantt-Diagramm für den beschriebenen Vorgang!

Erklären Sie Ihrem Tutor die Diagramme! Bitte bringen Sie die Diagramme in Papierform zum Tutoriumstermin mit.

Aufgabe 4.3: (15 Punkte)
Erklären Sie die Implementierung von Teilen des Windows-Schedulers!
- Eine neue Priorität wird für einen Thread mit Hilfe der Funktion `KiComputeNewPriority` berechnet. Erläutern Sie zeilenweise den genauen Ablauf der Funktion!
- Ein „Boost“ der Threadpriorität kann beispielsweise mit der Funktion `KiBoostPriorityThread` erzeugt werden. Erläutern Sie zeilenweise den genauen Ablauf der Funktion!

Aufgabe 4.4: (15 Punkte)
Erläutern Sie Ihrem Tutor folgende Begriffe und deren Verwendung im Kontext eines Betriebssystems!
- Virtueller Speicher
- Paging, Page-Fault, Page-Replacement Algorithm
- Swapping, Interne/Externe Fragmentierung

Aufgabe 4.5: (30 Punkte)
Implementieren Sie einen WRK-Systemaufruf, der es Ihnen ermöglicht, das verbleibende Quantum und das maximale Quantum für den aktuellen Thread zu bestimmen.

Der Systemruf soll die folgende Signatur haben:

```
NTSTATUS NtGetCurrentQuantumValues(
     PCHAR Quantum,
     PCHAR QuantumReset);
```

Die Werte `Quantum` und `QuantumReset` sind in der `KTHREAD` Datenstruktur zu finden. Überlegen Sie, wie sie diese Werte für den aktuellen Thread bestimmen können.

Folgen Sie bei der Implementierung den folgenden Schritten:

- Ergänzen Sie die Tabellen der Systemaufrufe und Argumente!
- Erstellen Sie eine Datei `quantuminfo.c` im Verzeichnis `\base\ntos\ex\`!
- Ergänzen Sie in der Datei `\base\ntos\ex\BUILD\makefile` die Zeile
  `$(OBJ)\quantuminfo.obj` (im Abschnitt `ccobjs=`)
  damit die neue Datei mit übersetzt wird!
- Implementieren Sie den Systemaufruf, inkludieren Sie die Datei `ntos.h`!
- Beachten Sie Parameterprüfung und Fehlerbehandlung!
- Kompilieren und Testen Sie Ihren erweiterten Kernel!

Implementieren Sie ein Testprogramm für Ihren Systemaufruf!

- Erstellen Sie eine User-Mode Bibliothek, die folgende Funktion bereitstellt und mit Hilfe des neuen Systemaufrufes implementiert:
  `BOOL GetQuantumInfo(PCHAR current, PCHAR max);`
- Erstellen Sie ein Programm, welches die User-Mode Bibliothek verwendet!

Erläutern Sie Ihrem Tutor Ihre Implementierungen (Kernel- und User-Mode) und führen Sie Ihr Programm vor!

Verpacken Sie Source-Code Dateien Ihres Testprogrammes (inklusive Makefile) und die ergänzte Datei quantuminfo.c in eine ZIP-Datei (blatt4.zip).
Diese ZIP-Datei muss mindestens 24 Stunden vor dem Tutoriumstermin in das Abgabesystem eingestellt werden.