Vorlesung Betriebssystemarchitektur WS 2007/08
Aufgabenblatt 5 vom 24. Januar 2008
(Vorstellung der Lösungen bei den Tutoren bis zum 06.02.2008)

Aufgabe 5.1: (15 Punkte)
Untersuchen Sie die Page-Replacement Algorithmen FIFO und LRU! Angenommen das System hat vier physikalische Speicherseiten, welche am Anfang leer sind. Zeichnen Sie jeweils ein Diagramm, welches die Verwendung des physikalischen Speichers für die unten angegebene Seitenzugriffsreihenfolge illustriert! Markieren Sie auftretende Page-Faults im Diagramm!

Reihenfolge der Seitenzugriffe:
1 2 3 1 4 5 1 2 3 4 5 5 1 5 2 3 1 4 5 1

Aufgabe 5.2: (5 Punkte)
Erläutern Sie *Belady's Anomaly* mit Hilfe der unten angegebenen Seitenzugriffsreihenfolge! Verwenden Sie drei bzw. vier physikalische Speicherseiten.

Reihenfolge der Seitenzugriffe:
1 2 3 4 1 2 5 1 2 3 4 5 1

Bitte bringen Sie die Diagramme (Aufgaben 5.1 und 5.2) in Papierform zum Tutoriumstermin mit und erläutern Sie sie Ihrem Tutor!

Aufgabe 5.3: (10 Punkte)
Gegeben sei ein 32-Bit-Rechnersystem. Die Seitenrahmengröße (Page-Frame Size) beträgt 1Kbyte. Das Betriebssystem unterstützt maximal 4Gbyte physikalischen Speicher. Die virtuelle Speicherverwaltung unterstützt lediglich eine Ebene von Page-Tables, d.h. zu jedem Prozess gehört genau eine Tabelle, die den gesamten virtuellen Speicher auf den physikalischen abbildet. Diese Tabellen können nicht ausgelagert werden. Der Rechner hat insgesamt 64Mbyte physikalischen Speicher. Das Betriebssystem benötigt mindestens 3Mbyte Speicher, der nicht ausgelagert werden kann.
Beantworten Sie Ihrem Tutor folgende Fragen:

- Wie viele Einträge enthält eine Page-Table?
- Wie viel physikalischer Speicher wird von einer Page-Table belegt? Jeder Page-Table Eintrag soll die Page-Frame Nummer und 2 Kontroll-Bits enthalten.
- Wenn zwei Prozesse in diesem System laufen, wie groß kann das Working-Set eines Prozesses maximal sein?

Aufgabe 5.4: (15 Punkte)
Erläutern Sie Ihrem Tutor die Rollen der *Standby-*, *Modified-*, *Free-* und *Zero page-* Listen in der Windows Speicherverwaltung! Welche Eigenschaften haben Seiten, die in den einzelnen Listen enthalten sind?

Aufgabe 5.5: (15 Punkte)
Beantworten Sie Ihrem Tutor folgende Fragen:

- Wie ist das Konzept der Zugriffskontrolle in Windows implementiert?
- Was ist ein *Security Descriptor*?
- Was ist der typische Aufbau einer DACL?
- Was ist der Zweck der SACL?

Aufgabe 5.6: (40 Punkte)
Das Programm `bmp_mandelbrot.c` (siehe Vorlesungsseite) berechnet ein fraktales Bild mit Hilfe einer Fraktal-Bibliothek. Die Bibliothek stellt folgende Funktion bereit, die in `algorithm.h` deklariert ist.

```
int getColorValuesAt(     double x,
                          double y,
                          unsigned char *red,
                          unsigned char *green,
                          unsigned char *blue)
```

Die Parameter `x` und `y` bezeichnen die Koordinaten eines Punktes innerhalb des komplexen Zahlenraums (siehe `bmp_fractal.c`). In den Parameter `red, green,` und `blue` werden die berechneten RGB-Werte (0-255) des Punktes zurückgeben.
Schreiben Sie in der Programmiersprache C ein Programm für Windows, welches das fraktale Bild (500x500 Pixel) im Bereich [-1,5…0,496;-1,0…0,996] mit Hilfe mehrerer Prozesse (1. Übergabeparameter **–p**) bzw. mit Hilfe mehrere Threads (1. Übergabeparameter **–t**) berechnen kann. Der 2. Übergabeparameter bezeichnet die Anzahl der zu verwendenden Berechnungseinheiten (Prozesse oder Threads). Die Ausgabe des Bildes soll in eine BMP-Datei erfolgen (3. Übergabeparameter).
Die berechneten Farbinformationen sollen in einen von allen Berechnungseinheiten gemeinsam genutzten Speicherbereich abgelegt werden und als BMP-Datei mit 24 Bit unkomprimierte Farbinformationen gespeichert werden. Die erzeugte BMP-Datei soll dem zur Verfügung gestellten Referenzdatei `mandelbrot.bmp` entsprechen
Eine detaillierte Beschreibung des BMP-Formates finden Sie auf der Vorlesungsseite.

Beispiel:

`>aufg56 -t 65 mandelbrot.bmp`	Das Programm startet 65 Threads für die Berechnung. Das Ergebnis wird in mandelbrot.bmp gespeichert.
`>aufg56 -p 65 mandelbrot.bmp`	Das Programm startet 65 weitere Prozesse für die Berechnung. Das Ergebnis wird in mandelbrot.bmp gespeichert.

Es sollen mindestens 65 Berechnungseinheiten unterstützt werden. Auf der Vorlesungsseite finden Sie die Fraktal-Bibliothek, welche mit Hilfe des `bmp_fractal` Programms die ebenfalls enthaltene BMP-Datei erzeugt.

Auftretende Fehler sollen durch die Ausgabe der Systemfehlermeldung, auf die Standardfehlerausgabe dem Benutzer angezeigt werden.

Erläutern Sie Ihrem Tutor Ihre Implementierung und führen Sie Ihr Programm vor!

Erstellen Sie ein Makefile für **`nmake`** mit dem Ziel **`aufg56.exe`**. Verpacken Sie alle Source-Code Dateien (`Fractal.lib` wird nicht benötigt) (inklusive Makefile) in eine ZIP-Datei (blatt5.zip). Kommentieren Sie ihren Quellcode!

Diese ZIP-Datei muss mindestens 24 Stunden vor dem Tutoriumstermin in das Abgabesystem eingestellt werden.