Vorlesung Betriebssysteme WS 2014/2015
Aufgabenblatt 2 vom 2.Dezember 2014
(Vorstellung der Lösungen bei den Tutoren bis zum 17.12.14)

Aufgabe 2.1: (20 Punkte)
Untersuchen Sie Eigenschaften des Betriebssystems Windows auf InstantLab mit Hilfe des Performance Monitors (perfmon)! Beobachten Sie die Zähler für `privileged time`, `interrupts/second`, `processor time` und `user time`!

- Geben Sie Auskunft über die aktuelle Messung. Welche Charakteristiken des Betriebssystems werden durch diese Werte erkennbar?
- Erklären Sie die Änderungen die durch das Ausführen einer Shell (cmd.exe), Maus- oder Tastaturereignisse auftreten. Versuchen Sie die gemessenen Werte zu maximieren.

Aufgabe 2.2: (10 Punkte)
Untersuchen Sie die Eigenschaften des Betriebssystem Linux! Lesen Sie dazu die Manualseiten: top(1) und ps(1)!

- Woran erkennt man die Prozesse die als Dämonen laufen?
- Wo lassen sich die von ps angezeigten Informationen im Dateisystem finden?

Aufgabe 2.3: (20 Punkte)
Kernel-Objekte repräsentieren Windows-Ressourcen, die von Applikationen verwendet werden können.

- Erläutern Sie für die Objekte File, Mutant, Directory und Event welche Betriebsmittel diese repräsentieren und welche Funktionalität sich dahinter verbirgt.
- Was ist ein Handle? Erläutern Sie, warum Handles nicht ohne weiteres über Prozessgrenzen hinaus verwendet werden können. Benennen Sie mindestens eine Windows-API-Funktion, mit der dies doch möglich ist.
- Der Kernel-Adressraum ist in zwei große Speicher-Bereiche unterteilt. Benennen Sie diese und erläutern Sie den Unterschied. Durch den Verbrauch welchen Speicher-Bereiches kann das System zum Quasi-Stillstand gebracht werden?
- Schauen Sie sich die Datei `handle.c` des Windows Research Kernels an. Ermitteln Sie durch Code-Review die maximale Anzahl von Betriebsmitteln, die ein Prozess besitzen kann. Benennen Sie außerdem die Konstante, die diesen Wert enthält.
- Nennen Sie 5 Aufgaben des Moduls Object Manager!

Aufgabe 2.4: (20 Punkte)
In dieser Aufgabe geht es um die Erzeugung von Prozessen. Machen Sie sich mit der Semantik des fork- und wait-Systemrufs vertraut. Betrachten Sie anschließend das folgende Programm. Welche Prozesse existieren zu den Zeitpunkten t={10s, 30s, …, 110s} im System?
Fertigen Sie einen Prozessgraph (Abbildung 1) an. Zur Vereinfachung wird angenommen, dass für die Systemrufe keinerlei Zeit beansprucht wird.

```
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

void main(void) {
    sleep(20);
    fork();
    sleep(20);
    fork();
    wait(NULL);
    sleep(20);
    fork();
    sleep(20);
}
```

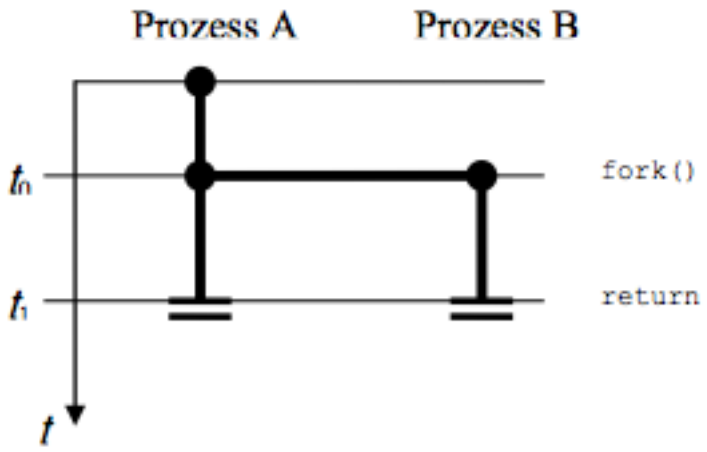

Abbildung 1: Prozessgraph

Bitte bringen Sie das Diagramm zum Tutoriumstermin mit und erläutern Sie es Ihrem Tutor!

Aufgabe 2.5: (30 Punkte)
Das Programm `bmp_mandelbrot.c` Windows / Linux (Programmrahmen auf der Vorlesungsseite) berechnet ein fraktales Bild als BMP-Datei mit Hilfe einer Fraktal-Bibliothek (x86|x86_64 für Windows und Linux). Die Bibliothek stellt folgende Funktion bereit, die in `algorithm.h` deklariert ist:

```
int getColorValuesAt(double x, double y,
                     BYTE *red, BYTE *green, BYTE *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 und Linux mit Hilfe der InstantLab Platform, welches das fraktale Bild (500x500 Pixel) im Bereich [-1,5…0,496;-1,0…0,996] mit Hilfe mehrerer Threads (1. Übergabe-Parameter) berechnen kann. Die Ausgabe des Bildes soll in eine BMP-Datei erfolgen (2. Übergabe-Parameter).

Die berechneten Farbinformationen sollen von allen Berechnungseinheiten in die BMP-Datei als 24 Bit unkomprimierte Farbinformationen gespeichert werden. Die erzeugte BMP-Datei soll der zur Verfügung gestellten Referenzdatei `mandelbrot.bmp` entsprechen.
Der initiale Thread/Prozess startet die Berechnungs-Threads und wartet auf deren Beendigung. Eine detaillierte Beschreibung des BMP-Formates finden Sie auf der Vorlesungsseite.

Beispiel:
```
>aufg25 65 mandelbrot.bmp
```

Das Programm startet 65 Threads 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, wenn verfügbar, auf die Standardfehlerausgabe dem Benutzer angezeigt werden.

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

Erstellen Sie ein Makefile-Regelsystem mit dem Ziel `aufg25`, welches unter Windows aufg25.exe und unter Linux aufg25 erzeugt. Verpacken Sie alle Quell-Code Dateien für Windows und Linux (inklusive Makefile-Regelsystem) in eine ZIP-Datei (blatt25.zip). Diese ZIP-Datei muss mindestens 24 Stunden vor dem Tutoriumstermin in das Abgabesystem eingestellt werden.