

Vorlesung Betriebssystemarchitektur WS 2004/05

Aufgabenblatt 4 vom 9. Dezember 2004

(Vorstellung der Lösungen bei den Tutoren bis zum 5. Januar 2005)

Aufgabe 4.1: (20 Punkte)

Die 100 Angestellten einer Firma sind in 4 gleich großen Abteilungen organisiert. Jeder Angestellte besitzt neben einer bestimmten Menge von Standard-Zugriffsrechten noch Abteilungsspezifische Zugriffsrechte. Des weiteren existieren 2 privilegierte Nutzer, die über zusätzliche Rechte verfügen sollen (z.B. Administratoren).

Es soll ein gemeinsamer Ordner für die Angestellten jeder Abteilung erzeugt werden, auf den die beiden privilegierten Nutzer jedoch keinen Zugriff haben sollen.

- Wie kann das beschriebene Szenario mit den Betriebssystemmitteln unter UNIX (rwx-Bits, Gruppen) umgesetzt werden?
- Wie mit Hilfe von Windows 2000/XP ?

Nach einer Firmenzusammenlegung sollen pro Abteilung 3 zusätzliche Nutzer mit Standard- und Abteilungsrechten angelegt werden, die zunächst jedoch keinen Zugriff auf den oben beschriebenen gemeinsamen Ordner der entsprechenden Abteilung haben dürfen.

- Wie kann dieses Ziel mit UNIX erreicht werden ?
- Wie mit Windows 2000/XP ?

Schreiben Sie Ihre Lösungsvorschläge in Stichpunkten auf und fertigen Sie eine Skizze an. Bringen Sie die Ergebnisse zum Tutoriums-Termin in Papierform mit.

Hinweis:

Für die Abgabe von Programmen gelten folgende Regeln:

- Die ZIP-Datei hat dem Namen aufg[Aufgabennummer].zip
- Die ZIP-Datei enthält ein Datei Makefile oder makefile im Wurzelverzeichnis des Archivs
- Der Make- Aufruf (nmake bzw. make) mit dem Zielnamen erzeugt diese Datei
 - z.B. Zielname=aufg42.exe Aufruf: nmake aufg42.exe Erzeugt: aufg42.exe
 - z.B. Zielname=auf42 Aufruf: make aufg42 Erzeugt: aufg42
- Zieldateien, die in der Aufgabenstellung angegeben sind, werden in dem gleichen Verzeichnis wie das Makefile erzeugt.
- Die ZIP-Datei muss als Attachment an die Mail-Adresse `bs@hpi.uni-potsdam.de` mindestens 24 Stunden werktags vor dem Tutoriumstermin geschickt werden
- Der Betreff der Mail folgt dem Schema: AUFGABE=[Aufgabennummer] GRUPPE=[Gruppennummer] + eventuelle Zusätze
- Die Aufgaben werden automatisch ausgewertet. Eine Statusmeldung wird nach erfolgter Auswertung als Email an den Absender geschickt.
- Stellen Sie sicher das Ihre Abgaben den Anforderungen entsprechen und kompilierbar sind!

Aufgabe 4.2: (30 Punkte)

Auf der Webseite <http://www.dcl.hpi.uni-potsdam.de/uebung/> befinden sich zwei C-Programme "lesen.c" und "schreiben.c", die zusammen ein kleines Chat-System bilden, mit dem man

innerhalb einer Windows-Domäne kommunizieren kann. Die Programme benutzen Win32-Mailslots. Die Programme benötigen als Übergabe-Parameter einen Chat-Raum-Namen.

- a. Informieren Sie sich in der MSDN-Hilfe über die Benutzung der Win32-API-Aufrufe *CreateMailslot*, *ReadFile*, *WaitForMultipleObjects*, *GetOverlappedResult*. Ändern Sie das Lese-Programm so ab, dass mehrere Chat-Räume gleichzeitig angezeigt werden können (es soll nur ein Thread laufen). Die Namen der Chat-Räume sollen auf der Kommandozeile angegeben werden. Das kompilierte Programm soll **aufg42a.exe** heißen.
- b. Schreiben Sie ein C-Programm, welches die Funktionen "lesen" und "schreiben" in einem Prozess vereint. Es können mehrere Threads benutzt werden. Das kompilierte Programm soll **aufg42b.exe** heißen.

Verpacken Sie die Source-Code-Dateien inklusive eines Makefiles in eine ZIP-Datei und schicken Sie diese ZIP-Datei als Attachment mindestens 24 Stunden vor dem Tutoriumstermin an `bs@hpi.uni-potsdam.de` mit dem Betreff `AUFGABE=4.2 GRUPPE=<Ihre Übungsgruppennummer>`

Die Mails werden automatisch bearbeitet: die ZIP-Datei wird ausgepackt und daraufhin überprüft, ob die Datei **Makefile** enthalten ist. Danach wird der Befehl **nmake aufg42a.exe** bzw. **nmake aufg42b.exe** ausgeführt.

Erklären Sie Ihrem Tutor die Programme und führen Sie diese vor.

Aufgabe 4.3: (30 Punkte)

Erweitern Sie das Programm aus Aufgabe 3.4 (`mandelbrot.c`) in der Weise, dass für die Berechnung der Teil-Fraktale Prozesse benutzt werden und die Ergebnisdaten in einem von den Prozessen gemeinsam genutzten Speicherbereich abgelegt werden. Aus Ergebnisdaten soll ein Bitmap Bild (BMP) mit 24 Bit unkomprimierten Farbinformationen für jeden Pixel erzeugt werden. Das Hauptprogramm, welches die Koordinierung der Prozesse übernimmt, soll als 1. Parameter die Anzahl der zu benutzenden Worker-Prozesse und als 2. Parameter den Dateinamen für das zu erzeugende Bild erwarten. Es sollen maximal 100 Worker-Prozesse gestartet werden können. Auf der Webseite <http://www.dcl.hpi.uni-potsdam.de/uebung> ist die Datei-Formatbeschreibung und ein Beispielprogramm für die zu erzeugende BMP-Bilddatei angegeben. Implementieren Sie Ihre Lösung in der Programmiersprache C entweder unter Windows 2000/XP (`CreateFileMapping()`, ...) oder unter Linux (`mmap()`, `shmget()`, ...) . Auftretende Fehler sollen behandelt und vorhandene Fehlerbeschreibung vom System dem Benutzer ausgegeben werden.

Verpacken Sie alle Source-Code-Dateien inklusive eines Makefiles in eine ZIP-Datei und schicken Sie diese ZIP-Datei als Attachment mindestens 24 Stunden werktags vor dem Tutoriumstermin an `bs@hpi.uni-potsdam.de` mit dem Betreff

`AUFGABE=4.3 GRUPPE=<Ihre Übungsgruppennummer> OS=WIN32` oder

`AUFGABE=4.3 GRUPPE=<Ihre Übungsgruppennummer> OS=LINUX` je nach Betriebssystem.

Die Mail wird automatisch bearbeitet: die ZIP-Datei überprüft, ob die Datei **Makefile** im lokalen Verzeichnis enthalten ist. Danach wird der Befehl **nmake aufg43.exe** (Windows) bzw. **make aufg43** (Linux) ausgeführt, welcher das Hauptprogramm (**aufg43.exe** bzw. **aufg43**) im lokalen Verzeichnis und eventuelle Zusatzprogramme erzeugen muss. Beachten Sie auch die Hinweise auf diesem Übungsblatt.

Erklären Sie Ihrem Tutor das Programm und führen Sie es vor. Beantworten Sie dabei folgende Fragen:

- Wie haben Sie die Arbeitsaufteilung zwischen den Arbeits-Prozessen realisiert?
- Wie haben Sie den gemeinsamen Speicherzugriff für die Prozesse realisiert?
- Wie erzeugen Sie die Bitmap Datei?