



Betriebssysteme: Konzepte, Dienste, Schnittstellen (Betriebssysteme und betriebssystemnahe Programmierung)

SS 2005 – Übungsblatt 13 (optional)

Ausgabe: 18. Juli 2005

Abgabe: bis spätestens 25. Juli 2005
im Fachschaftsraum Mathematik
oder per email an c.markmann@uni-wuppertal.de

Aufgabe 1. *Kindprozesse/-threads*

Auf welche Art unterscheidet sich das Erzeugen von Kindprozessen mittels `vfork()` beziehungsweise mittels `fork()`?

Wie erzeugt man in einem Prozeß einen Thread?

Wie unterscheidet sich ein Kindprozeß von einem Kindthread?

Was ist ein Mutex und wozu dient er?

Für welche Einsatzzwecke sind Prozesse, für welche Threads geeigneter?

Aufgabe 2. *atexit(),...*

Erläutern Sie jeweils kurz einen sinnvollen Anwendungszweck der folgenden „Funktionen“:

`atexit()`, `pthread_mutex_lock()`, `system()`, `WTERMSIG()`,
`signal()`, `sigprocmask()`, `execlp()`, `read()`, `perror()`,
`stty()`.

Aufgabe 3. *Plattenorganisation*

Erläutern Sie die folgenden Begriffe aus dem Umfeld des Festplattenlayouts:

- Partitionstabelle

- Filesystem
- aktive Partition
- erweiterte Partition
- FAT32
- low-level-Format
- Journal
- sparse File

Aufgabe 4. *Aufgaben eines Betriebssystems*

Welche Aufgabenbereiche hat ein modernes Betriebssystem abzudecken? Nennen und diskutieren Sie mindestens fünf verschiedene Aufgabenbereiche.

Aufgabe 5. *System-Calls*

Wie unterscheiden sich System-Calls von normalen Unterprogrammaufrufen?

Welche Funktionalität bieten System-Calls an?

Welches Vorgehen ist bei dem Aufruf von System-Calls empfehlenswert?

Was ist unter dem Begriff ILP64 zu verstehen?