Übungen zur Vorlesung
Betriebssysteme II
Sommersemester 2002

AG BS
Jan Peleska, Aliki Tsiolakis

Blatt 2

Analyse des Linux-Schedulers und der Prozesserzeugung mit fork

Aufgabe 1: Prozesstabelle

Erläutern Sie den Zusammenhang zwischen den Datenstrukturen

- Verkettete Liste auf der Grundlage der Struktur `struct task_struct`
- Array `pidhash`
- Array `task`
- Array `tarray_freelist`

Aufgabe 2: PID

Unter welchen Bedingungen kann eine PID mehrfach vergeben weden?
Wie werden dann die Prozesse noch eindeutig zugeordnet (vgl. z.B. Umgang mit PID in `signal.c`)?

Aufgabe 3: PID und fork()

Beschreiben Sie die Strategie, nach der für den Kindprozess die PID ausgewählt wird. Können Sie hierfür eine formale Spezifikation als Prädikat über Prozesstabelle und über die globalen Hilfsvariablen `next_safe` etc. angeben? (`fork.c`, Funktion `get_pid()`)

Aufgabe 4: `schedule()`

Wie wird ein neuer rechenbereiter Prozess in die `runqueue` eingeordnet?

Aufgabe 5: Scheduler Queues

Finden Sie heraus, in welchen Situationen gleichzeitig (z.B. von System call, scheduler und/oder Interrupt Handler) an bestimmten Queues geändert/gelesen wird. Warum ist dies konsistent?

Aufgabe 6: Kommentierung von `schedule()`

Kommentieren Sie den Code von `schedule()`, so dass die entscheidenden Scheduling-Entscheidungen transparent (d.h. für sterbliche Kreaturen verständlich) werden.

Abgabe: Bis Montag, 27. Mai 2002, vor dem Tutorium.

Geben Sie für alle Aufgaben eine **schriftliche Lösung** ab. Diese soll für Aufgabe 6 auch der von Ihnen dokumentierte Sourcecode (also u.U. auch nur Ausschnitte aus den Kernel-Source-Dateien) beinhalten. Bitte schicken Sie **zusätzlich** ein Archiv aller relevanten Dateien per Email an `tsio@informatik.uni-bremen.de`.