

3. Assignment

Process scheduling

Issue: May 11—Due: May 18

Exercise 5: UNIX process scheduling.

10 Points

The process scheduler used in BSD UNIX utilises a priority scheduling algorithm (see lecture): Initially, each process starts with a *base priority* in the range $[-20, 20]$, which on average is 0¹. Based on system load, the scheduler may change the priority of a process. The process with the highest priority is scheduled next.

If a process has used few CPU time recently, its priority is increased. The scheduling algorithm quickly forgets CPU consumption in order to favour processes with alternating CPU workload: The algorithm forgets 90% of the most recently used CPU time within $5 \times \text{load}$ seconds, where *load* is the average number of processes in *ready* state during the last few minutes².

Note the distinction between CPU-consumption, and other things a process might do while being executed, e.g., performing I/O operations.

Discuss the following issues:

1. Are CPU-intensive processes preferred if the system is under high load, or if the system is under low load?

*CPU-intensive Prozesse werden bevorzugt, falls das System unter niedrigem Load steht. Dies kann man sich anhand der "Vergessensrate" der zuletzt genutzten CPU-Zeit verdeutlichen: Falls der Load hoch ist, so wird der Ausdruck "5 * load" auch groß. D.h. es dauert lange bis die genutzte CPU-Zeit des CPU-intensiven Prozesses vergessen wird. Falls der Load des Systems niedrig ist, so ist der Ausdruck "5 * load" klein, und der Scheduler vergisst die zuletzt genutzte CPU-Zeit recht schnell. Da die zuletzt genutzte CPU-Zeit zur Priorisierung von Prozessen benutzt wird, ist es also für einen CPU-intensiven Prozess entscheidend wie schnell diese vom System wieder vergessen wird. Denn je schneller die zuletzt genutzte CPU-Zeit vergessen wird, desto schneller wird der Prozess wieder höher priorisiert. (Group 6)*

Short: *load hoch \Rightarrow 5 x load gross \Rightarrow CPU-Zeit wird langsamer vergessen (90% in 5 x load Sek.) \Rightarrow nur langsame Prioritaetserhoehung fuer CPU-intensive Prozesse \Rightarrow CPU-intensive Prozesse nicht bevorteilt.*

2. How is an I/O-intensive process scheduled after it is done with its I/O operation?

Recent cpu usage is quite low when a process is waiting for I/O (I/O is mostly done by I/O processors not consuming any CPU time). Hence, after the I/O event, the processes priority is increased and the process is likely to be executed next. (Group 11)

Short: *Prozess ist I/O-intensiv \Rightarrow Prozess verbrauchte in naher Vergangenheit wenig CPU-Zeit (I/O von I/O-Prozessoren ausgefuehrt, nicht CPU) \Rightarrow protokollierte CPU-Zeit fuer Prozess gering \Rightarrow Prioritaet hoch \Rightarrow nach Beendigung von I/O Prozess schnell geschcheduled.*

3. Is it possible that a very CPU-intensive process repeatedly becomes postponed for ever?

Mit dem oben genannten Scheduling Algorithmus wird verhindert, dass CPU intensive Prozesse verhungern (starvation), da diese im Laufe der Zeit eine höhere Priorität bekommen als sie eigentlich aufgrund ihrer CPU Auslastung bekommen würden (das Vergessen von 90% der

¹See the `nice(1)` command which allows a user to be nice, i.e., to start a process with reduced priority.

²The `top(1)` command displays load averages, priority, and nice values of all processes.

CPU Zeit implementiert ein oft als 'aging' bezeichnetes Verhalten). Dadurch werden auch diese Prozesse irgendwann berücksichtigt, da ihre Priorität immer weiter ansteigt. (Group 3)

Short: *CPU-intensiver Prozess \Rightarrow verbrauchte CPU-Zeit hoch \Rightarrow Priorität sinkt in Relation zu anderen Prozessen \Rightarrow Prozess wird unterbrochen \Rightarrow Scheduler "vergisst" seine CPU-Zeit (90% in 5 x load Sek.) \Rightarrow Priorität steigt wieder an \Rightarrow beizeiten Prozess wieder auf CPU.*

4. When the system load increases, what effect do CPU-intensive processes have on the interactive response times of the system?

Durch die Adaption der Vergessensrate der zuletzt genutzten CPU-Zeit an den aktuellen Load, haben CPU-intensive Prozesse keinen bemerkenswerten Einfluss auf die interaktive Response-Time des Systems. Sobald nämlich der Load steigt, dauert es immer länger bis der Scheduler die zuletzt genutzte CPU-Zeit vergisst und werden CPU-intensive Prozesse in Relation immer niedriger priorisiert. Die interaktive Response-Time wird aber durch I/O-intensive Prozesse bestimmt (z.B. Shell), schließlich werden Programme, die auf I/O (bspw. Mausbewegungen, Tastaturanschläge) warten, vom Scheduler in Relation priorisiert, da sie wenig CPU Zeit zuletzt genutzt haben. Gerade durch die Einbeziehung des Load-Wertes in die Berechnung der Vergessensrate, wird die interaktive Response-Time nicht durch einen hohen Load nennenswert beeinflusst. (Group 6)

Short: *load steigt \Rightarrow 5 x load gross \Rightarrow CPU-Zeit wird langsamer vergessen (90% in 5 x load Sek.) \Rightarrow nur langsame Prioritätserhöhung fuer CPU-intensive Prozesse \Rightarrow Priorität interaktiver Prozesse (CPU-arme Prozesse, meiste Zeit wird auf Input-Event [Tastendruck, Mausbewegung] gewartet) steigt in Relation zu anderen Prozessen \Rightarrow interaktive Prozesse auch unter hohem load reaktiv.*

Send your answers in a plain text email to Alexander and Stefan³. If you insist on attachments, have an eye on UTF-8 encoding and UNIX end-of-line conventions of the attached plain text file.

³alexander.holupirek@uni-konstanz.de, stefan.klinger@uni-konstanz.de