Advanced Operating Systems
Report 5

37186305
航空宇宙工学専攻修士一年
荒居秀尚

2018 年 11 月 21 日

1 The difference between nice priorities and RT priorities

1.1 Description of processes

Every process can be categorized into three types.

Interactive process
Those processes which are categorized in this type always need to be interactive with the user. The biggest feature of this type is that processes need to wait long time for the action of the user such as keyboard input or mouse manipulation. These processes need to wake up immediately when the input comes, in order to prevent user from feeling slow. Examples of this type of processes are command shell, text editor, and GUI applications.

Batch process
This type of processes don't need the manipulation of the user, and they are often running as background processes. Responsiveness is not important for this type, so they are sometimes degraded their priorities by the scheduler. Example of this type of processes are compilers, search engine for the database, and scientific calculation.

Real Time process
The scheduling requirements are very strict for this type and they should not be disturbed by low priority processes. They need to function quickly and stably. Example of this type of processes are video or audio applications, robot control, and sensing program.

1.2 Scheduling policy

Each Linux process has its scheduling policy.Real Time processes have three options, SCHED_FIFO, SCHED_RR, and SCHED_DEADLINE.

SCHED_FIFO

FIFO is the abbreviation of ”First In First Out”.When the scheduler allocated CPU to the process which has this policy, the position of process descriptor in the execution queue would not change.If there aren’t any process which has higher priority than the current process (with this policy), the current process will not stop using the CPU even if there is a process which has the same priority as the current process.

SCHED_RR

RR is the abbreviation of ”Round Robbin”.When the scheduler allocated CPU to the process which has this policy, the position of process descriptor will be moved to the end of the execution queue.When the process run out of the allocated quantum, the process stop running and wait until the other processes which have the same priorities as the current one and have the policy SCHED_RR use one quantum.

SCHED_DEADLINE Process with this policy has the deadline and the priority value is set to the highest automatically.

1.3 Scheduling for Real Time process

Every Real Time process has its own RT priority.This priority is a value between 1 and 99.The higher this value is, the lower its priority.The scheduler always pick the process which has the highest priority, which means that execution of those processes which has lower priority than the RT process are suppressed while RT processes are executed.

RT processes give over the execution right to another process only when the incident below happens.

- The process was preempted by another process which has higher real time priority.
- The process interrupted the execution and stopped running with the status of TASK_INTERRUPTIBLE or TASK_UNINTERRUPTIBLE.
- The process was interrupted and the status changed to TASK_STOPPED or TASK_TRACED, or the process was forced to exit and the status changed to EXIT_ZONBIE or EXIT_DEAD.
- The process called the system call sched_yield() and spontaneously give over the execution right.
- The process has the attribute of SCHED_RR and run out of the quantum.