RTS Project 1

Part I: Linux Kernel Building

Part II: Linux Scheduling Policy Analysis

Advisor: Prof. Tei-Wei Kuo

TA: Wen Sheng Lim
d08922028@csie.ntu.edu.tw

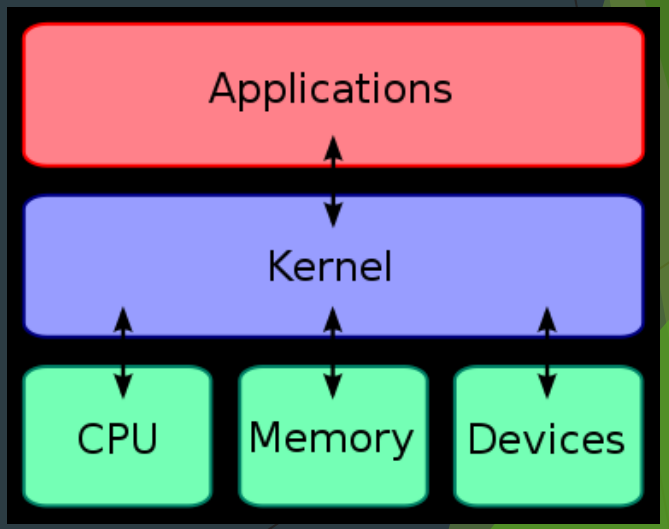
Slides credit: Han-Yi Lin

Outline

- What is “Kernel”
- Environment Setup
- Build Linux Kernel
- Linux Scheduling Policy Analysis
- Project Requirements
- Submission Rules

What is “Kernel”?

- The kernel[1] is a fundamental part of a modern computer’s operating system.
- The kernel's primary functions are to
  - Manage the computer's hardware and resources
    - E.g., CPU, main memory, I/O devices, and so on.
  - Allow applications to run and use these resources



Environment Setup

- Oracle VM VirtualBox[2]
  - https://www.virtualbox.org/wiki/Downloads
- Ubuntu 20.04 64bits LTS (or newer) [3]
  - https://ubuntu.com/download/desktop
- Install the Ubuntu on the VirtualBox
  - Recommended: memory>=4GB, disk>=60GB, assign 1 CPU for VM to avoid complexity of multi-core scheduling

Build Linux Kernel (1/5)

- After the installation, please login Ubuntu and open a terminal to start building your Linux kernel[4]

```
rts2022@rts2022-VirtualBox:~$ uname -a
Linux rts2022-VirtualBox 5.15.0-52-generic #58~20.04.1-Ubuntu SMP
 Thu Oct 13 13:09:46 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
rts2022@rts2022-VirtualBox:~$
```

Build Linux Kernel (2/5)

- $ sudo apt-get install build-essential libncurses5-dev flex bison libssl-dev libelf-dev dwarves zstd mpv
- $ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.15.71.tar.xz
- $ tar xvf linux-5.15.71.tar.xz
- $ cd linux-5.15.71
- $ make mrproper

Build Linux Kernel (3/5)

- $ make menuconfig
  - Make CONFIG_SYSTEM_TRUSTED_KEYS and CONFIG_SYSTEM_REVOCATION_KEYS empty
  - Disable CONFIG_DEBUG_INFO_BTF
- $ make
  - You can use make -j# (# is the number of your physical cores) to create multiple threads to speed up the kernel building
- $ sudo make modules_install

Build Linux Kernel (4/5)

- $ sudo make install
- $ sudoedit /etc/default/grub
  - Configure the following 2 items
    - GRUB_TIMEOUT_STYLE=menu
    - GRUB_TIMEOUT=10
- $ sudo update-grub
- $ sudo reboot

Build Linux Kernel (5/5)

- Now, you can select the kernel you installed in the GNU grub to boot your Ubuntu.

- Then, you can use terminal and type “uname -a” to check the kernel version.

```
rts2022@rts2022-VirtualBox:~$ uname -a
Linux rts2022-VirtualBox 5.15.71 #4 SMP Fri Oct 21 23:34:30 CST
 2022 x86_64 x86_64 x86_64 GNU/Linux
```

References

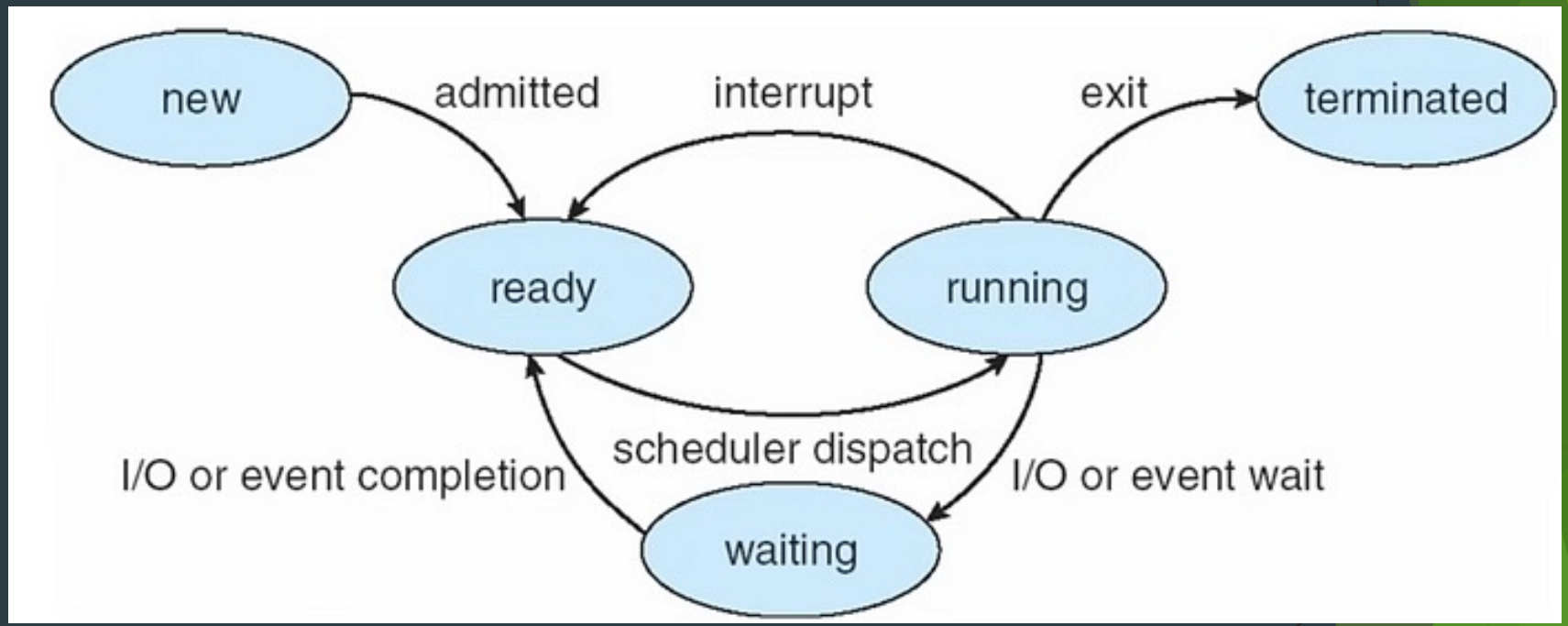
- [1] Wikipedia https://en.wikipedia.org/wiki/Kernel_(operating_system)
- [2] Oracle VM VirtualBox https://www.virtualbox.org/
- [3] Ubuntu https://ubuntu.com/
- [4] Linux Kernel in a Nutshell http://www.kroah.com/lkn/

SCHEDULING IN LINUX (補充資料)

Process Life Cycle

- A process is not always ready to run.
- The scheduler must know the status of every process in the system when switching between tasks.
- A process may have one of the following states:
  - Running — The process is executing at the moment.
  - Waiting — The process is able to run but is not allowed to because the CPU is allocated to another process. The scheduler can select the process at the next task switch.
  - Sleeping — The process is sleeping and cannot run because it is waiting for an external event. The scheduler cannot select the process at the next task switch.
- The system saves all processes in a process table.

Transitions between Process States



The Need of the Scheduler

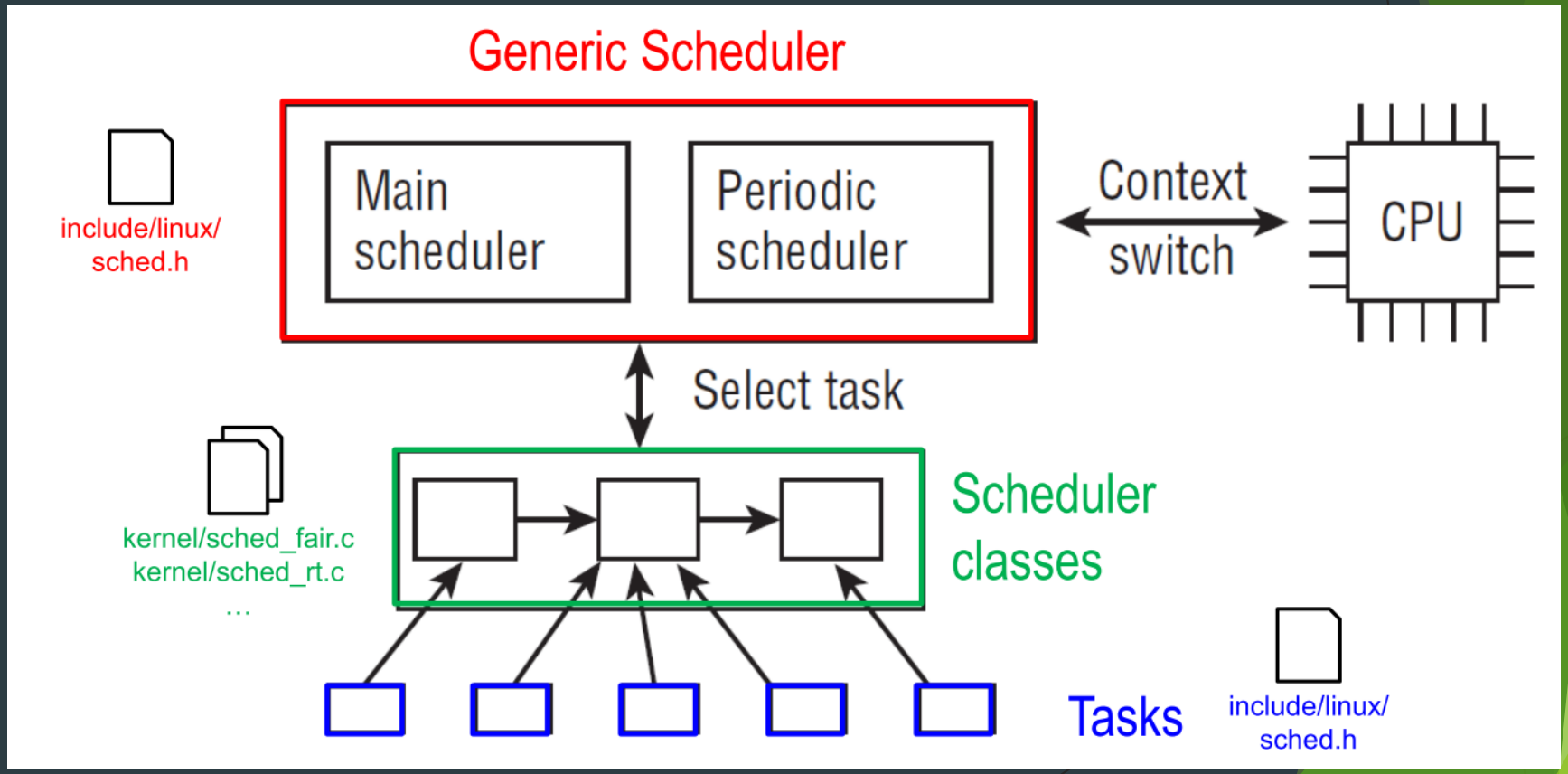
- A unique description of each process is held in memory and is linked with other processes by means of several structures.
- This is the situation facing the scheduler, whose task is *to share CPU time between the programs to create the illusion of concurrent execution*.
- Scheduler responsibilities —
  - Scheduling policy
  - Context switching

Linux Scheduling Subsystem

Generic Scheduler

Scheduler Classes

Task | Task | Task



Scheduling in Linux (1/2)

- The schedule function is the starting point to an understanding of scheduling operations.
- It is defined in “kernel/sched/core.c” and is one of the most frequently invoked functions in the kernel code.
- Not only priority scheduling but also two other soft real-time policies required by the POSIX standard are implemented.
- E.g., completely fair scheduling, real-time scheduling and scheduling of the idle task, etc.

Scheduling in Linux (2/2)

- The scheduler uses a series of data structures to sort and manage the processes in the system.
- Scheduling can be activated in two ways:
  - Main scheduler: Either directly if a task goes to sleep or wants to yield the CPU for other reasons,
  - Periodic scheduler: Or by a periodic mechanism that is run with constant frequency to check from time to time if switching tasks is necessary
- Generic scheduler = Main + Periodic schedulers

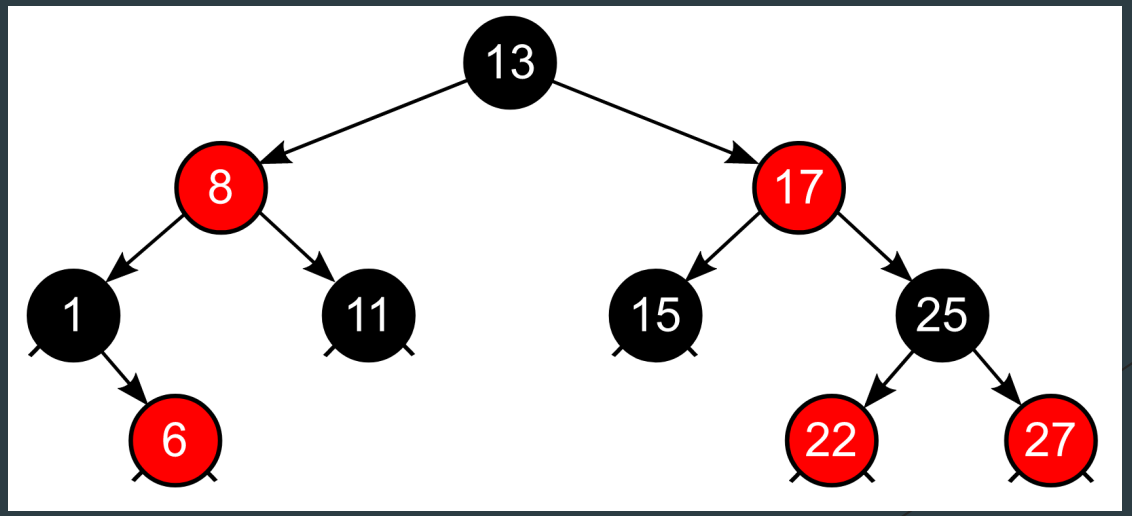
Linux Scheduling Policy Testing and Analysis

Linux Scheduling Policy Testing and Analysis

- Linux Scheduling Policy Classes
  - Normal Scheduling policies (Non-real-time)
    - SCHED_OTHER, SCHED_BATCH, SCHED_IDLE.
  - Real-Time policies
    - SCHED_FIFO, SCHED_RR, SCHED_DEADLINE.
- The default scheduling policy is non-real-time.
- In this part, using Linux deadline scheduling policy (EDF) to schedule threads in a process.

The Linux Deadline Scheduler

- Earliest Deadline First (EDF)
  - Each task is tracked as a node of a red-black tree
  - Nodes are ordered by dynamic scheduling deadline



Using Real-time Schedulers

- Play a video along with a CPU-intensive application
  - **CPU-intensive application (e.g., compression)**
    - $ sudo apt-get install p7zip-full sysstat
    - $ 7z b 100 -md16
  - **Video (**with a media player**)**
    - $ mpv --loop=inf video.mp4
    - Choose a suitable video that can be played in your VM and consumes some CPU resources
- Apply real-time scheduling for all threads and CPU-intensive threads
  - **Example:** $ sudo chrt --all-tasks --deadline --sched-runtime 700000 --sched-deadline 10000000 --sched-period 10000000 -p 0 $(pgrep mpv)
  - Observe and analysis dropped frames in different configurations

```
AO: [pulse] 44100Hz stereo 2ch float
VO: [sdl] 640x480 yuv420p
AV: 00:01:26 / 00:03:09 (45%) A-V:  0.394 Dropped: 12
```

Print Information for Real-Time Tasks

- Modify the kernel to print a message whenever scheduling parameters are changed (e.g., by chrt) for a task
- $ sudo dmesg

```
[10701.608517] SCHED_DEADLINE parameters set on pid 3792: runtime=700000, deadline=10000000, period=1000
0000
[10701.608526] SCHED_DEADLINE parameters set on pid 3794: runtime=700000, deadline=10000000, period=1000
0000
[10701.608532] SCHED_DEADLINE parameters set on pid 3795: runtime=700000, deadline=10000000, period=1000
0000
```

Hint

- The deadline scheduler is implemented in kernel/sched/deadline.c
- chrt uses sched_setattr() system call to assign the scheduler and scheduling parameters (runtime, deadline, period)
  - Parameters are passed via struct sched_attr
- The deadline scheduler maintains both assigned and runtime parameters
- The permission to assign a real-time scheduler
- Incremental kernel builds should be fast (a few minutes)

Project1 Requirements

- A PDF report within 2 pages and modified kernel source files
- Report should included: Linux Scheduling Policy Testing and Analysis and short description on kernel compilation steps (and problems, if any) and code tracing on deadline scheduler
  - For scheduling policies analysis, please compare the dropped frame among (1) different scheduling policies and (2) different parameters and priority combination
  - Describe WHAT you observe and WHY it behave like that
- Please show your names and student IDs in your report
- Describe work done by each member
- Be packed as one file named "RTS_PJ1_Team##_v#.zip"
  - RTS2024_PJ1_report.pdf
  - linux-5.15.71/kernel/sched/deadline.c

Submission Rules

- Project deadline: 2025/05/06 23:59
  - Delayed penalty: -20/Day
- Upload the zip file via NTU COOL
- DO NOT COPY THE HOMEWORK
  - Discussions among teams are encouraged, as long as properly credited

References

- Professional Linux® Kernel Architecture, Wolfgang Mauerer, Wiley Publishing, Inc.
- Deadline Task Scheduling https://www.kernel.org/doc/html/latest/scheduler/sched-deadline.html