

Jin-Soo Kim  
([jinsoo.kim@snu.ac.kr](mailto:jinsoo.kim@snu.ac.kr))

Systems Software &  
Architecture Lab.

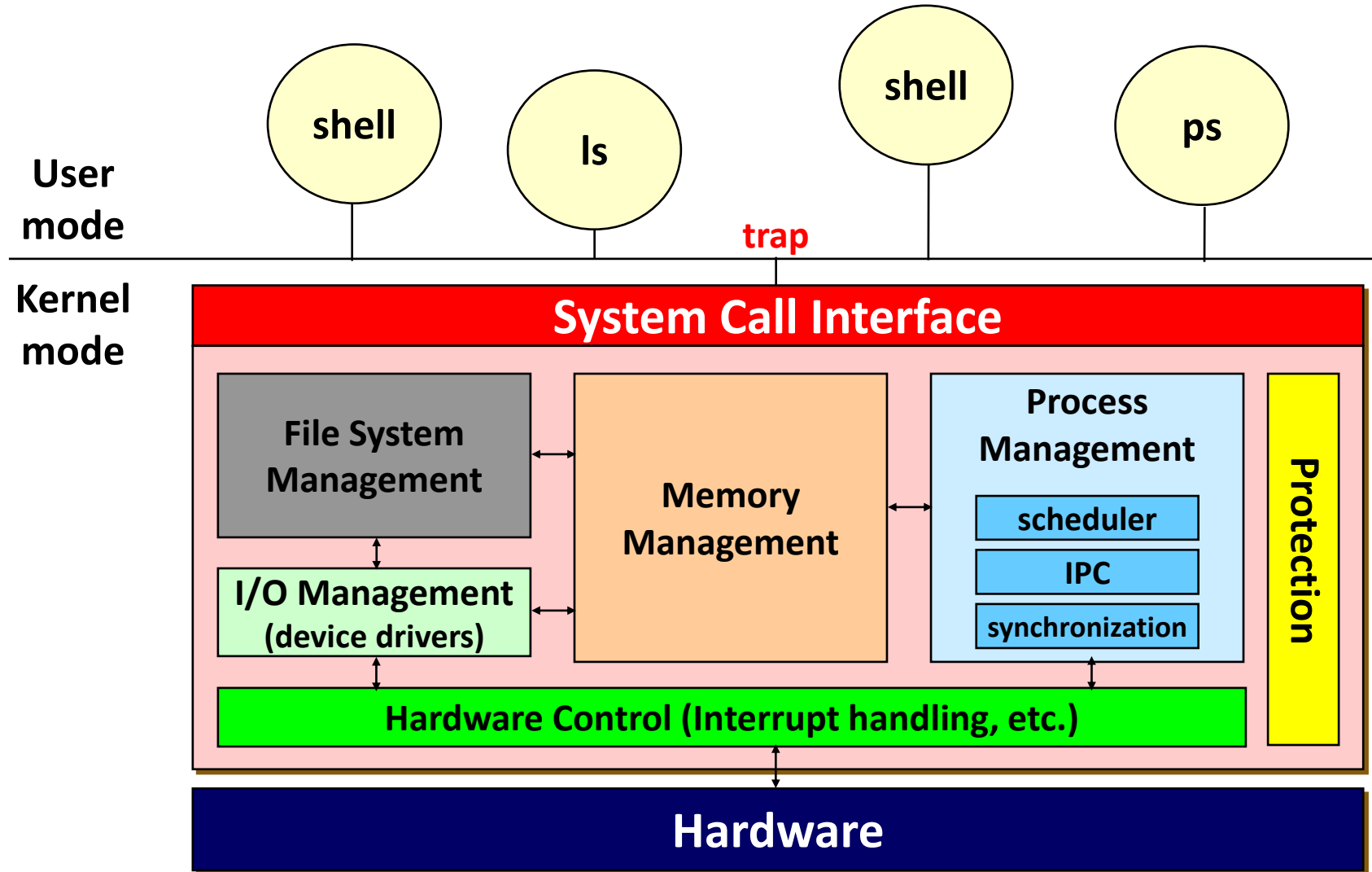
Seoul National University

Spring 2023

# Introduction to Operating Systems



# Operating System Internals



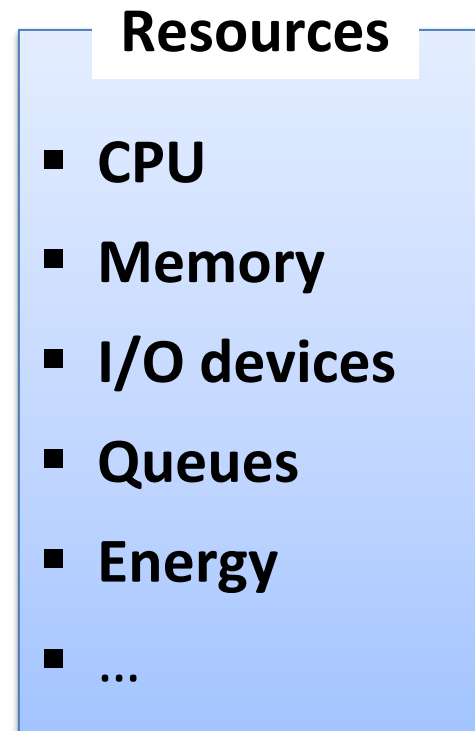
# OS: Application View

- OS provides an execution environment for running programs
- OS provides a(an) \_\_\_\_\_ view of the underlying computer system
  - What are the correct abstractions?
  - How much of hardware should be exposed?
- Typical OS abstractions
  - Processors → Processes, Threads
  - Memory → Address space (virtual memory)
  - Storage → Volumes, Directories, Files
  - I/O Devices → Files (+ ioctls)
  - Networks → Files (sockets, pipes, ...)



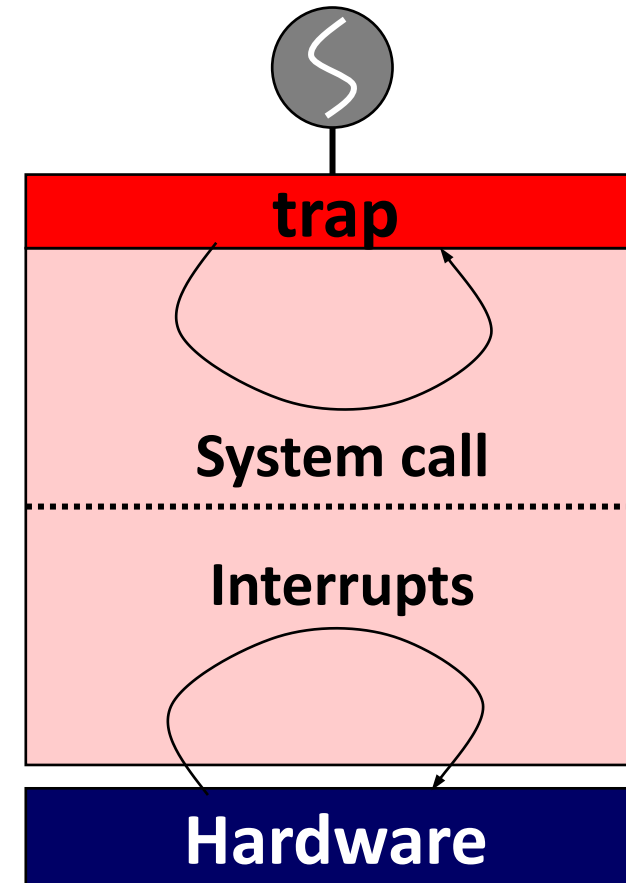
# OS: System View

- OS manages various resources of a computer system
- Sharing
- \_\_\_\_\_
- Fairness
- Efficiency
- ...



# OS: Implementation View

- OS is highly-concurrent, \_\_\_\_\_ software
- Two kinds of events
  - System calls
  - Interrupts

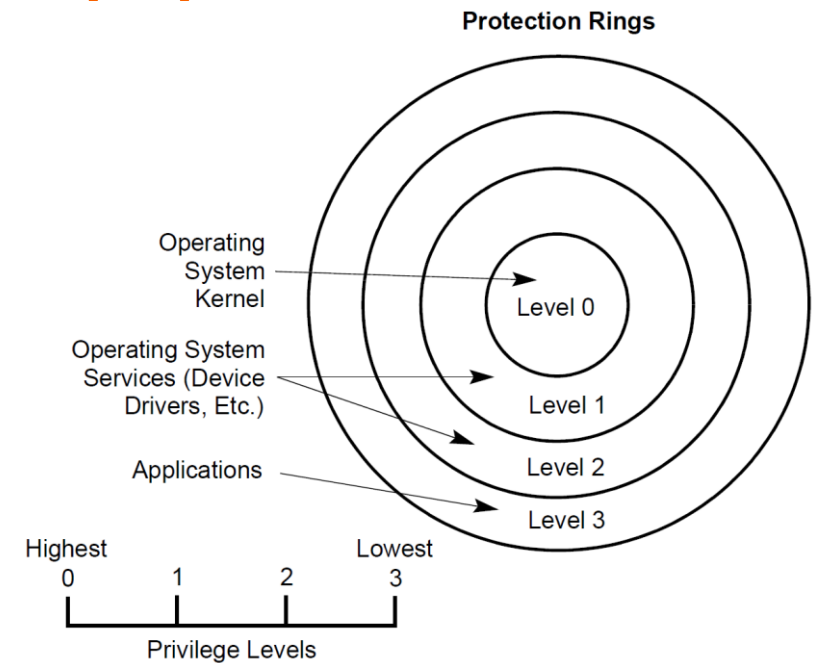


# Unix Features

- **Process control**
  - `fork()`, `exec()`, `wait()`, `exit()`
  - Pipes for inter-process communication (IPC)
- **Hierarchical file systems**
  - Special files: uniform I/O, naming, and protection
  - Removable file systems via `mount/umount`
  - i-node
- **Signals**
- **Shells**
  - Standard I/O and I/O redirection, filters
  - Shell scripts

# Architectural Support for OS (I)

- CPU modes of operation: kernel vs. user
  - 4 levels in x86: Ring 0 > 1 > 2 > 3
  - 3 levels in RISC-V: Machine > Supervisor > User
- Protected or unprotected instructions
  - Direct I/O access (e.g., in/out instructions in x86)
  - Accessing system registers
  - Memory state management
  - ...



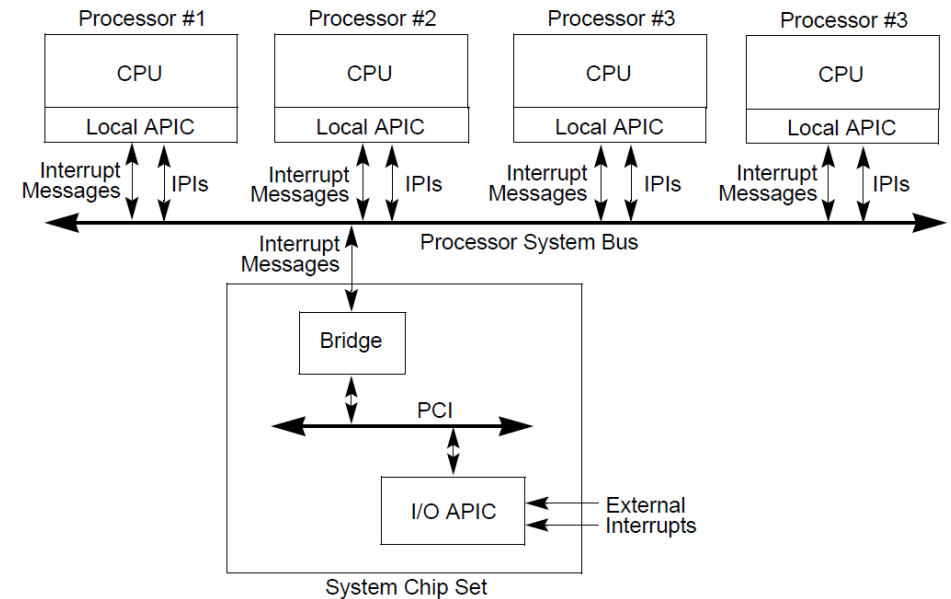
# Architectural Support for OS (2)

## ■ Interrupts

- Generated by hardware devices
- External interrupts vs. IPIs
- Asynchronous

## ■ Exceptions

- Generated by software executing instructions
  - **Faults** (unintentional, but possibly recoverable): page faults, protection faults, ...
  - **Traps** (intentional): `syscall` instruction in x86\_64 or `ecall` instruction in RISC-V
  - **Aborts** (unintentional and unrecoverable): parity error, machine error, ...
- Synchronous
- Exception handling is logically same as interrupt handling





# Architectural Support for OS (3)

- **Memory protection**
  - Segmentation
  - Paging
- **Timer**
- **DMA (Direct Memory Access)**
- **Atomic instructions**
  - Atomic inc/dec
  - Test-and-Set
  - Compare-and-Swap
  - LL (Load Locked) & SC (Store Conditional)
  - ...

# Assignment #1: Installing Ubuntu

- Install the latest VirtualBox 7.0.6 on your x86\_64 machine
  - <https://virtualbox.org>
- Install Ubuntu 22.04 LTS in your VirtualBox
  - <https://ubuntu.com>



- Download the reference Linux kernel 6.1.14

```
$ wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.1.14.tar.xz
```

```
$ tar xvf linux-6.1.14.tar.xz
```

# Assignment #1: Compiling Linux Kernel

- Install prerequisite packages

```
$ sudo apt install build-essential ncurses-dev libssl-dev flex bison libelf-dev zstd dwarves
```

- Configure the kernel

```
$ cd linux-6.1.14  
$ cp /boot/config-$(uname -r) .config  
$ scripts/config --disable SYSTEM_TRUSTED_KEYS  
$ scripts/config --disable SYSTEM_REVOCATION_KEYS
```

- Compile and install the kernel

```
$ make all -j 4  
$ sudo make INSTALL_MOD_STRIP=1 modules_install  
$ sudo make install
```

# Assignment #1: Modifying Linux Kernel

- Print your name and student ID during the booting
- Take a screenshot
- Be creative!



- Also take a screenshot of the following command:

```
jinsoo@xi:~$ cat /proc/version
Linux version 6.1.14 (jinsoo@xi) (gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0, GNU ld
(GNU Binutils for Ubuntu) 2.38) #1 SMP PREEMPT_DYNAMIC Mon Mar  6 11:17:30 KST 2023
```