

# Outline

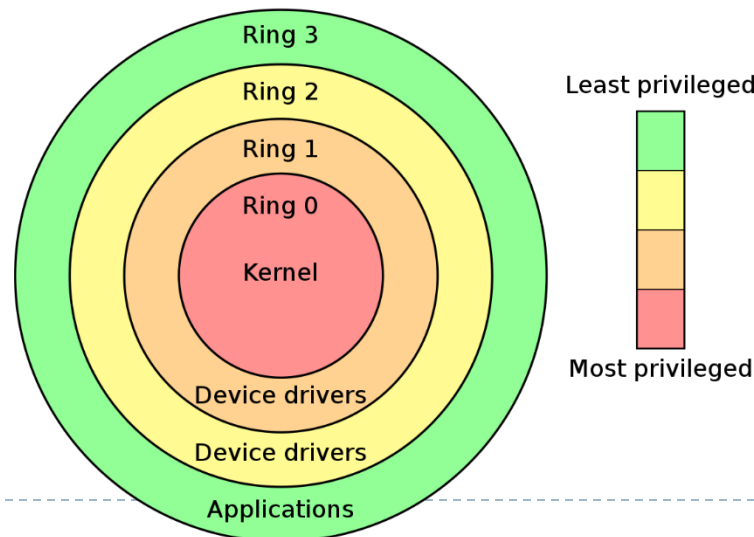
---

- ▶ **Security Protection Stages in OS**
  - ▶ Authentication
  - ▶ Authorization with Access Control
  - ▶ Logging, Monitoring & Auditing
- ▶ **Privilege Management in OS**

# Privileged Rings Inside OS

## Operating modes

- ▶ Kernel mode has the highest privilege, running the critical functions and services; user mode has the least privilege.
- ▶ Entities in the higher rings cannot call the functions and access the objects in the lower rings directly.
- ▶ Context switch is required to achieve the above procedure, system call, interrupt, etc.
- ▶ Status flag allows system to work in different modes.



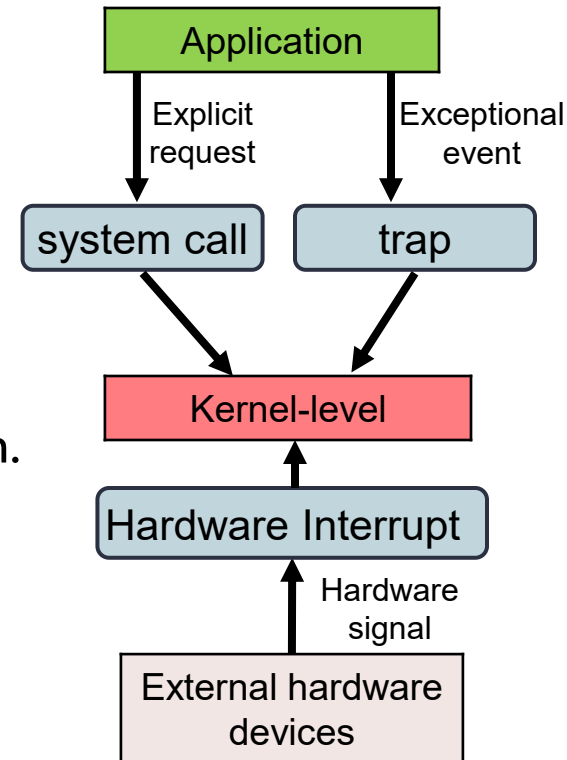
# Context Switch

## Different events can trigger the transition from user to kernel levels

- ▶ System call: user application explicitly makes a request to kernel for privileged operations
- ▶ Trap: user application gets an exceptional event or error and requests the kernel to handle.
- ▶ System call and trap belong to software interrupts,
- ▶ Hardware interrupt: hardware issues a signal to the CPU to indicate an event needs immediate attention.

## Switch procedure

- ▶ CPU stores process's states, and switches to the kernel mode by setting the status flag.
- ▶ Kernel handles the interrupt based on the interrupt vector in an interrupt table.
- ▶ CPU switches back to user mode and restores states



# How System Call is Issued and Handled

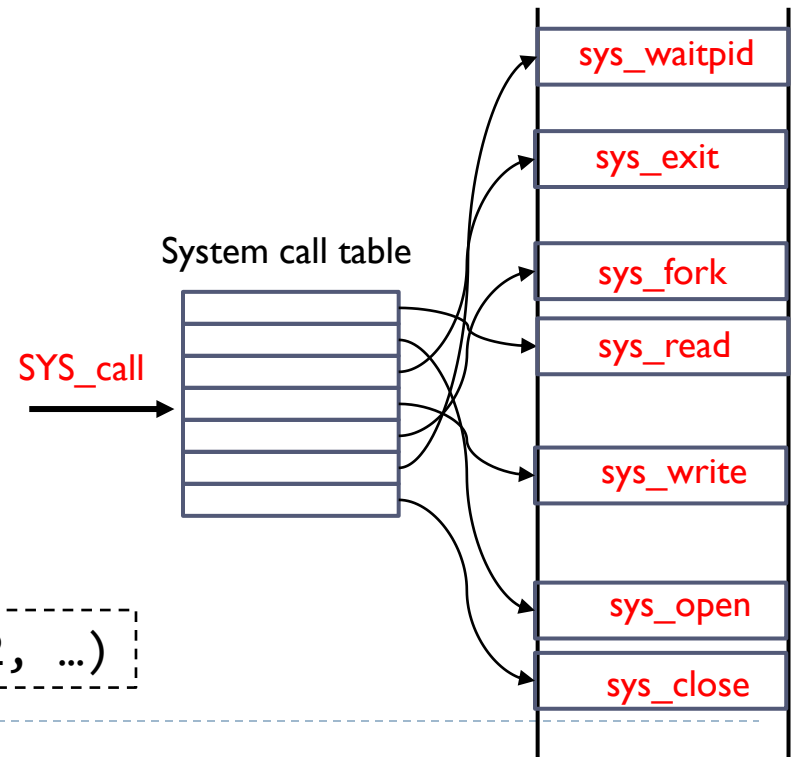
A system call is an interface that allows a user-level process to request functions or services from the kernel level.

- ▶ Process control
- ▶ File management
- ▶ Device management

## How to issue a system call?

- ▶ System call table: a table of pointers in the kernel region, to different system call functions.
- ▶ A user process passes the index of the system call and parameters with the following API:

```
syscall(SYS_call, arg1, arg2, ...)
```



# Rootkit

---

## Malware that obtains root privileges to compromise the computer

- ▶ Root user does not go through any security checks, and can perform any actions to the system
  - Insert and execute arbitrary malicious code in the system's code path
  - Hide its existence, e.g., malicious process, files, network sockets, from being detected.

## How can the attacker gain the root privileges?

- ▶ Vulnerabilities in the software stack: buffer overflow, format string...

There are some common techniques for rootkits to compromise the systems.