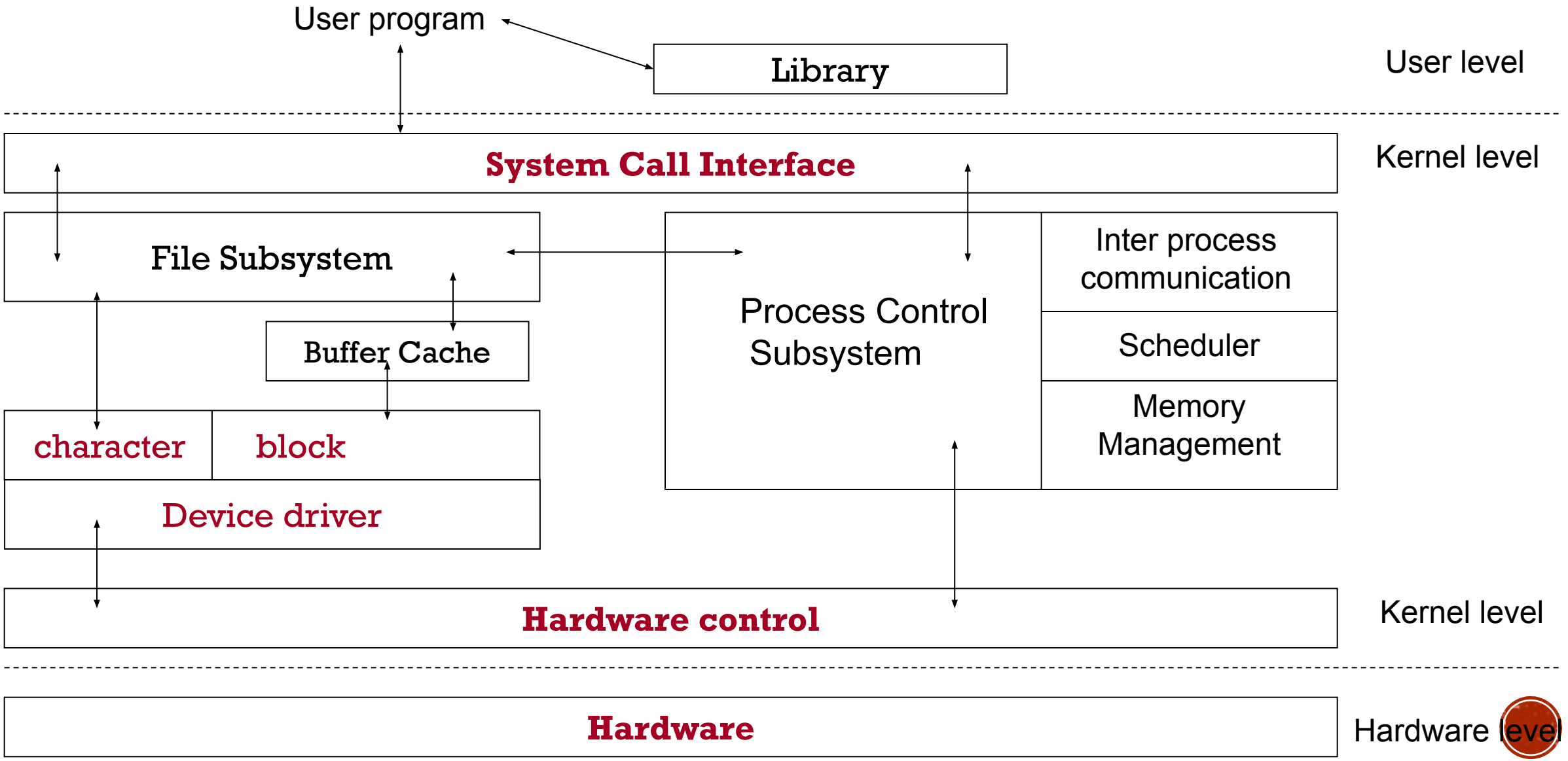
INTRODUCTION TO THE KERNEL

KERNEL ARCHITECTURE

KERNEL ARCHITECTURE (UNIX)



KERNEL ARCHITECTURE

- The block diagram of the kernel shows various modules and their relationships to each other.
- The file system and the process system are the two major components of the kernel.
- The architecture shows three levels:
  - User
  - Kernel
  - Hardware

KERNEL ARCHITECTURE

- System calls look like ordinary function calls in C programs.
- Libraries maps these function calls to the primitive needed to enter the OS.
- Assembly language programs may invoke system calls directly without a system call library.
- Programs frequently use other libraries such as standard I/O library(stdio.h) to provide a more sophisticated use of the system calls.
- Libraries are linked with the programs at compile time and are thus part of the user programs.

KERNEL ARCHITECTURE

- The set of system calls are partitioned into two:
  - System calls those interact with the file subsystem.
  - System calls those interact with the process control subsystem.
- The file subsystem
  - manages files,
  - allocating file space, and monitoring free space,
  - controlling access to files and
  - retrieving data for users.

KERNEL ARCHITECTURE

- The processes interact with the file subsystem via a specific set of system calls, such as
  - open, creat,
  - close, lseek
  - read,
  - write,
  - stat (query the attributes of a file),
  - chown (change the record of who owns the file),
  - chmod (change the access permission of a file).
  - chgrp (change the group ownership of a file).

KERNEL ARCHITECTURE

- The file subsystem accesses file data using a buffering mechanism that regulates data flow between the kernel and secondary storage devices – Buffer Cache.
  - The kernel attempts **to minimize the frequency of disk access** by keeping a pool of **internal data buffers, called the buffer cache, that contains the recently used disk blocks content.**
- The buffering mechanism interacts with block I/O device drivers to initiate data transfer to and from the kernel.
- Device drivers are the kernel modules that control the operation of peripheral devices.

KERNEL ARCHITECTURE

- Block I/O devices are random access storage devices.
  - E.g. a tape driver may allow the kernel to treat a tape unit as a random access storage devices.
- The file subsystem also interacts directly with raw (character oriented) I/O device drivers without intervention of a buffering mechanism.

KERNEL ARCHITECTURE

- The process control subsystem is responsible for
  - process synchronization,
  - inter-process communication,
  - memory management, and
  - process scheduling.
- The file and process control subsystems interact when loading a file into memory for execution.
- The process subsystem reads executable files into memory before executing them.

KERNEL ARCHITECTURE

- Some of the system calls for controlling processes are
  - fork,
  - exec (overlay the image of a program onto the running process),
  - exit,
  - wait (synchronize process execution with the exit of a previously format process),
  - brk (control the size of memory allocated to a process),
  - signals (control process response to extraordinary events).

KERNEL ARCHITECTURE

- The **memory management module** controls the allocation of memory
- If at any time the system does not have enough physical memory for all processes, the kernel moves them between main memory and secondary memory so that all processes get a fair chance to execute.
- It is done by swapper.
  - Swapper process is sometimes called as the scheduler, because **it schedules the allocation of memory for processes** and influence the operation of the CPU scheduler.

KERNEL ARCHITECTURE

- The **CPU scheduler module** allocates the CPU to processes:
  - It schedules them to run in turn until they voluntarily relinquish the CPU while awaiting a resource or until the kernel preempts them when their recent runtime exceeds a **time quantum**.
  - The CPU scheduler then chooses the highest **priority** eligible process to run; the original process will run again when it is the highest priority eligible process available.

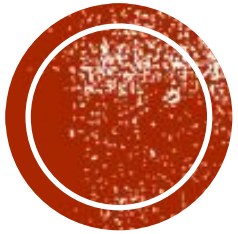
- There are several forms of **inter-process communication**
  - Ranging from asynchronous **signalin**g of events to synchronous transmission of **messages** between the processes.

KERNEL ARCHITECTURE

- The hardware control is responsible for handling interrupts and for communicating with the machine.
- Devices such as disks or terminals may interrupt the CPU while a process is executing.
  - If so, the kernel may resume execution of the interrupted process after servicing the interrupt.
    - Interrupts are not serviced by special processes but by special functions in the kernel, called in the context of the currently running process.

KERNEL DATA STRUCTURES

- Most kernel data structures occupy fixed-size tables rather than dynamic allocation space.
- The advantage of this approach is, the kernel code is simple.
  - But it limits the number of entries for a data structure and reports an error to the user, if the kernel runs out of entries
- The simplicity of the kernel algorithms is considered more important here.
  - The algorithms, that use simple loops to find free table entries, that is sometimes more efficient than complicated allocation schemes.



THANK YOU !