



# Operating System Overview



# Process Management

- ▶ A process is a program in execution. It is a unit of work within the system. Program is a **passive entity**, process is an **active entity**.
- ▶ Process needs resources to accomplish its task
  - ▶ CPU, memory, I/O, files
  - ▶ Initialization data
- ▶ Process termination requires reclaim of any reusable resources
- ▶ Single-threaded process has one **program counter** specifying location of next instruction to execute
  - ▶ Process executes instructions sequentially, one at a time, until completion
- ▶ Multi-threaded process has one program counter per thread
- ▶ Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - ▶ Concurrency by multiplexing the CPUs among the processes / threads



# Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- ▶ Creating and deleting both user and system processes
- ▶ Suspending and resuming processes
- ▶ Providing mechanisms for process synchronization
- ▶ Providing mechanisms for process communication
- ▶ Providing mechanisms for deadlock handling



# Memory Management

- ▶ To execute a program all (or part) of the instructions must be in memory
- ▶ All (or part) of the data that is needed by the program must be in memory.
- ▶ Memory management determines what is in memory and when
  - ▶ Optimizing CPU utilization and computer response to users
- ▶ Memory management activities
  - ▶ Keeping track of which parts of memory are currently being used and by whom
  - ▶ Deciding which processes (or parts thereof) and data to move into and out of memory
  - ▶ Allocating and deallocating memory space as needed



# Storage Management

- ▶ OS provides uniform, logical view of information storage
  - ▶ Abstracts physical properties to logical storage unit - **file**
  - ▶ Each medium is controlled by device (i.e., disk drive, tape drive)
    - ▶ Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- ▶ File-System management
  - ▶ Files usually organized into directories
  - ▶ Access control on most systems to determine who can access what
  - ▶ OS activities include
    - ▶ Creating and deleting files and directories
    - ▶ Primitives to manipulate files and directories
    - ▶ Mapping files onto secondary storage
    - ▶ Backup files onto stable (non-volatile) storage media



# Mass-Storage Management

- ▶ Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- ▶ Proper management is of central importance
- ▶ Entire speed of computer operation hinges on disk subsystem and its algorithms
- ▶ OS activities
  - ▶ Free-space management
  - ▶ Storage allocation
  - ▶ Disk scheduling
- ▶ Some storage need not be fast
  - ▶ Tertiary storage includes optical storage, magnetic tape
  - ▶ Still must be managed – by OS or applications
  - ▶ Varies between WORM (write-once, read-many-times) and RW (read-write)



# Performance of Various Levels of Storage

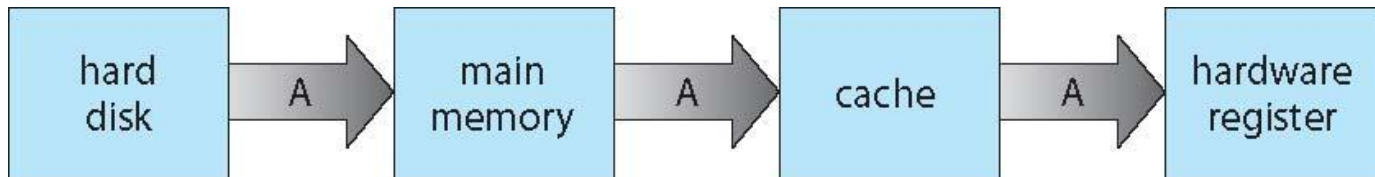
Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit



# Migration of data “A” from Disk to Register

- ▶ Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- ▶ Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache
- ▶ Distributed environment situation even more complex
  - ▶ Several copies of a datum can exist
  - ▶ Various solutions covered in Chapter 17





# I/O Subsystem

- ▶ One purpose of OS is to hide peculiarities of hardware devices from the user
- ▶ I/O subsystem responsible for
  - ▶ Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - ▶ General device-driver interface
  - ▶ Drivers for specific hardware devices



# Protection and Security

- ▶ **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
  - ▶ **Security** – defense of the system against internal and external attacks
    - ▶ Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
  - ▶ Systems generally first distinguish among users, to determine who can do what
    - ▶ User identities (**user IDs**, security IDs) include name and associated number, one per user
    - ▶ User ID then associated with all files, processes of that user to determine access control
    - ▶ Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
    - ▶ **Privilege escalation** allows user to change to effective ID with more rights
- 

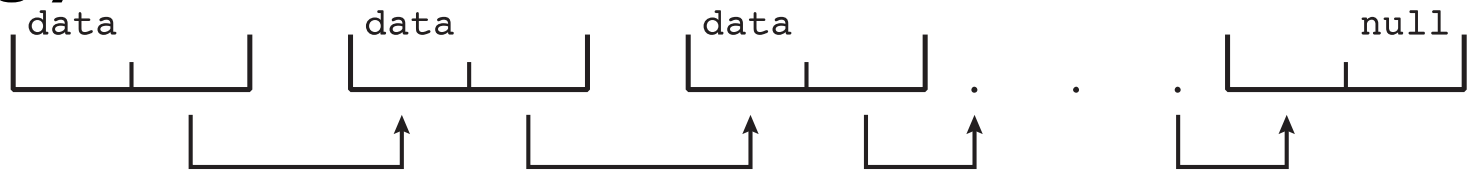


# Kernel Data Structures

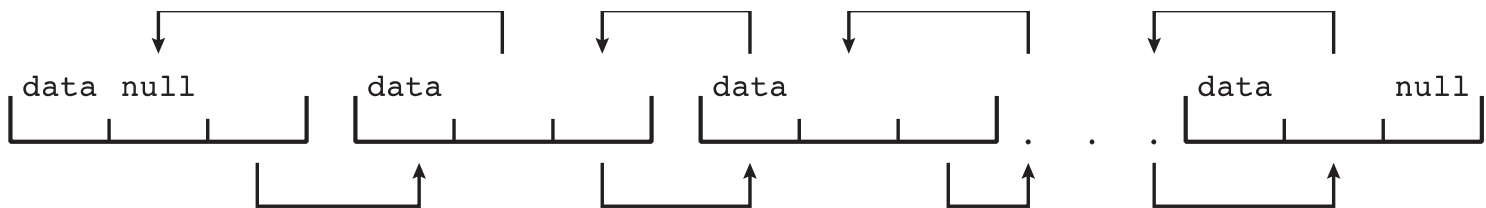
---

- Many similar to standard programming data structures

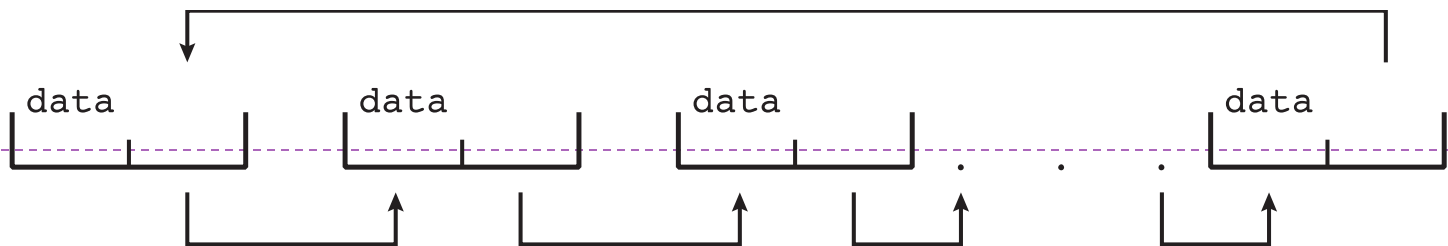
- ***Singly linked list***



- ***Doubly linked list***



- ***Circular linked list***



# Kernel Data Structures

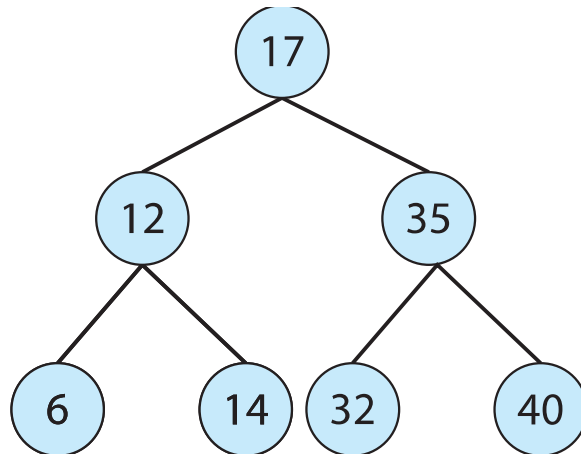
---

- ▶ **Binary search tree**

left  $\leq$  right

- ▶ Search performance is  $O(n)$

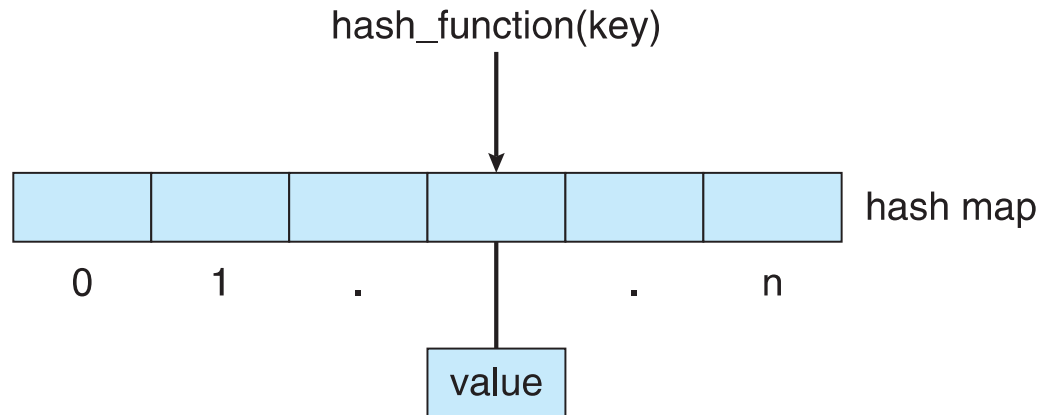
- ▶ **Balanced binary search tree** is  $O(\lg n)$



# Kernel Data Structures

---

- ▶ **Hash function** can create a **hash map**



- ▶ **Bitmap** – string of  $n$  binary digits representing the status of  $n$  items
- ▶ Linux data structures defined in  
**include files** `<linux/list.h>`, `<linux/kfifo.h>`,  
`<linux/rbtree.h>`



# Summary

---

- ▶ Process Management
- ▶ Memory Management
- ▶ Storage Management
- ▶ Protection and Security



# Check your understanding

---

- ▶ What are the Memory management activities ?
- ▶ List the OS activities in mass storage Management
- ▶ What are the frequently used kernel data structures?

