Lecture 2

What we have learnt

- What is system
- OS definition and its goals
- System components :processors, memory, storage , I/O subsystems, busses , files & network
- Abstract view of computing systems
- Operating system as a resource manager

User View of OS

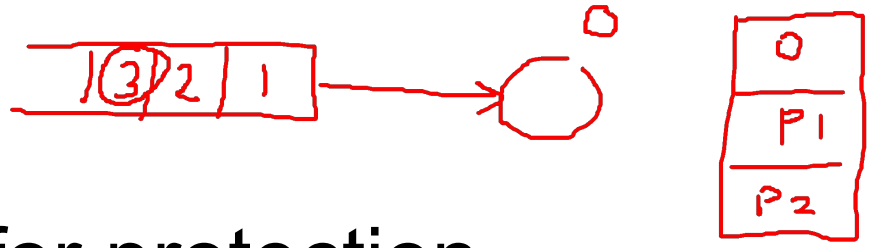
- Provides ease of operation
- Supports resource sharing
  - Sequential sharing : A resource is allocated for exclusive use of program eg. CPU
  - Concurrent sharing: Two or more program concurrently use same resource eg. Files, memory, disk array etc.

Systems View of OS

- OS acts as Resource allocator and is responsible for handling resource request, resource allocation and optimal utilization of system resources.
- Acts as control program that manages the execution of user program and prevents error and improper use of computer system.

What OS Does?

- Maintain a list of authorized users
- Construct list of all resources in the system
- Initiate execution of programs
- Maintain resource usage information by programs and current status of programs
- Maintain current status of all resources and allocate resources to programs when requested
- Perform scheduling
- Maintain information for protection



Goals Of Operating System

- Provide user interface for ***ease of operation***
  - Command line interface (CLI) eg. Unix, DOS
  - Graphical User Interface (GUI) eg. Windows
- ***Efficient*** use of hardware & software resources (system)
- Maximize ***System performance***
- Protection and access control
- Ability to evolve and offer new services
- Footprint of OS should be small !!!

Can we measure System Efficiency, Performance and user convenience ?

- Efficiency---- CPU efficiency
- System performance ----- Throughput
- User services ---- Turn around time ,
  Response time

Operating System and Computer Architecture are tightly bonded together.

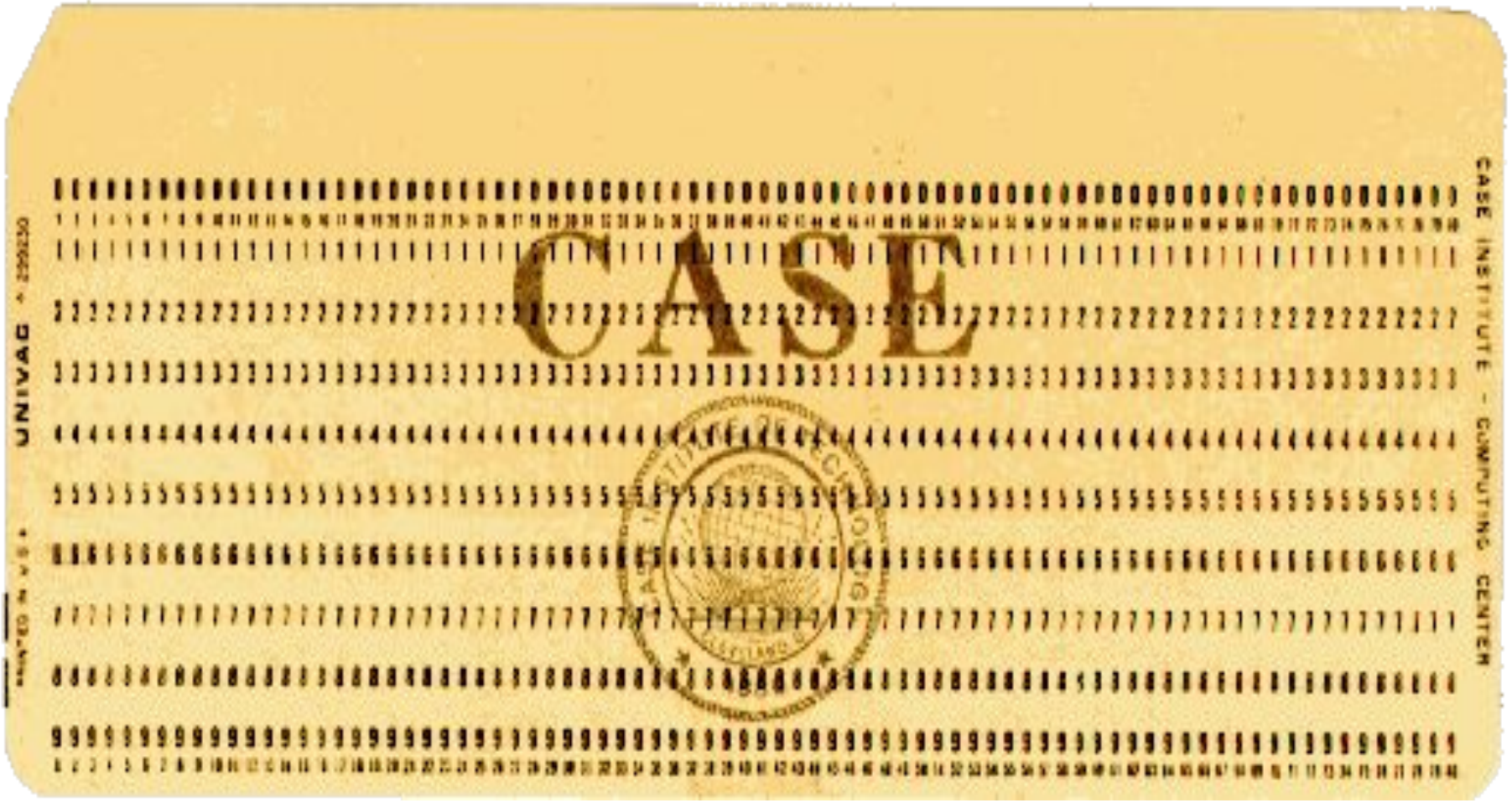
- Change in architecture usually leads to change in OS.
- The requirements of OS have led to several architectural changes.

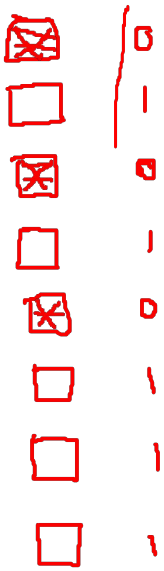
Different Computer Systems

Mainframe Systems

- Are large systems
- Are very expensive
- Early mainframes were Batch Systems
- They used punched cards and tape for data and code input
- They used printer for results and error reporting

Punched Card(Hollerith's card)





Mainframe Systems

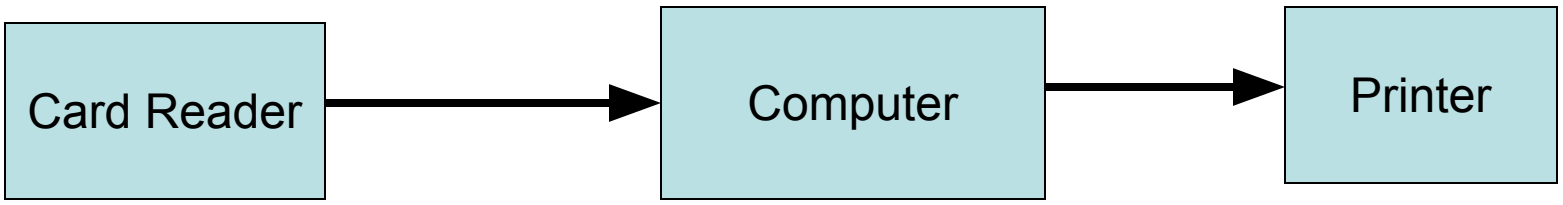
- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers back to monitor

Batch Systems

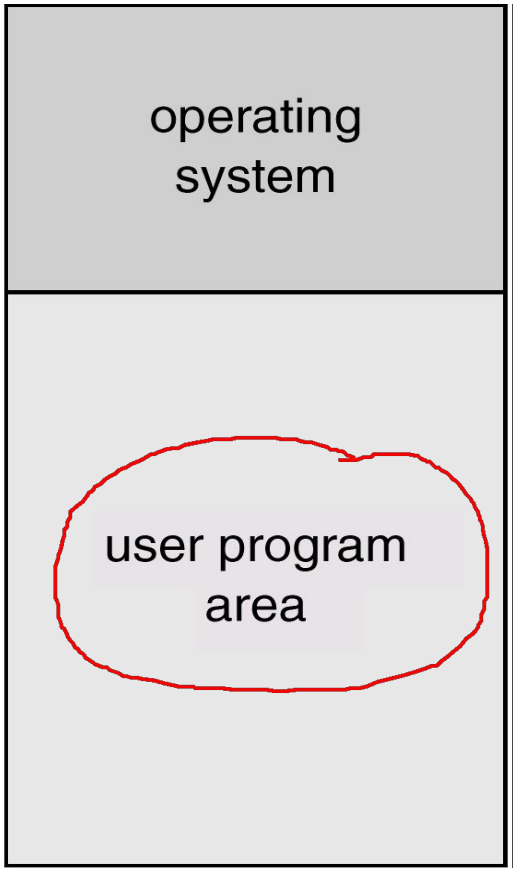
- Users submits job
- Operator forms a batch by bunching user jobs
- Loads job into the card reader
- Computer executes these jobs sequentially and produces results on printer
- Printed output is collected by operator and distributed to users.

Characteristics of Batch System

- Once a job starts executing , until finished no other job can execute
- Low CPU utilization because when program is reading data or producing output CPU is idle
- No user interaction
- Batch systems are appropriate for executing large jobs that require little interaction



Batch system Memory Layout



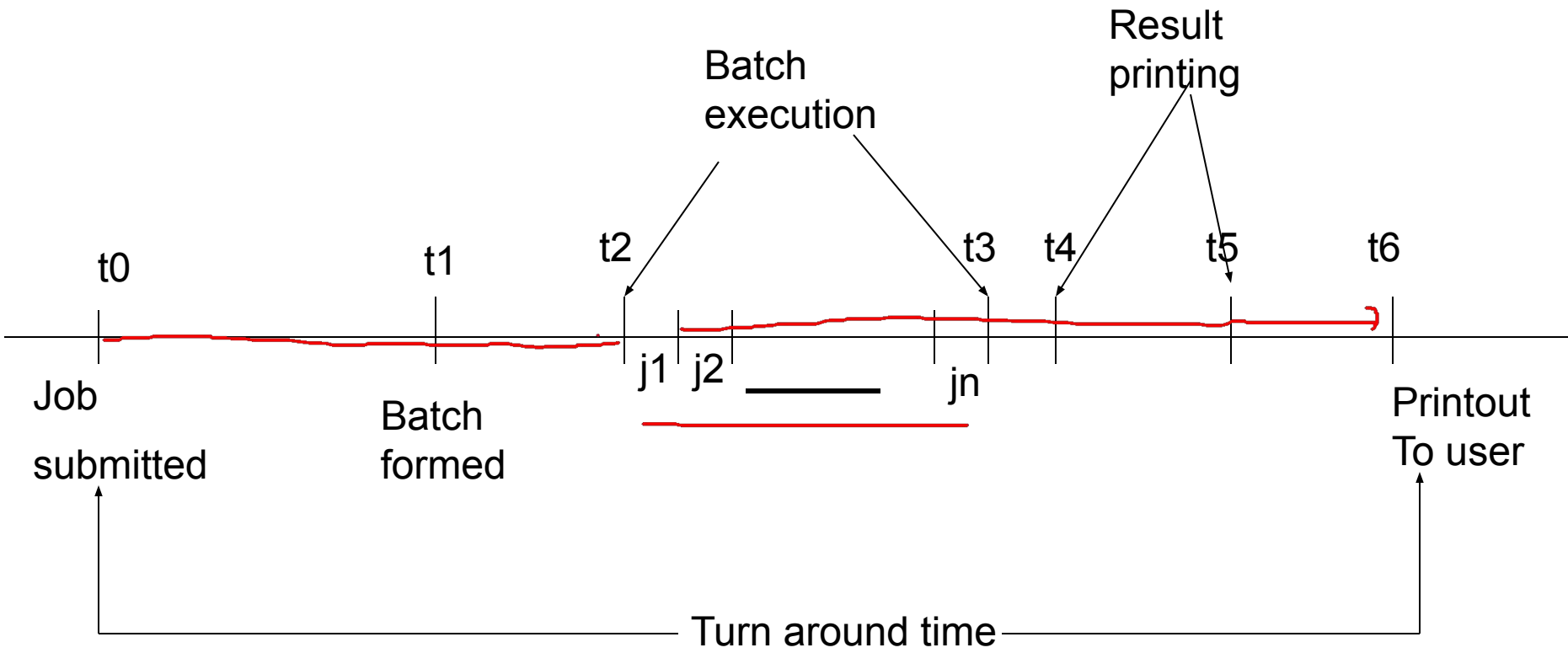
- Main task of the Batch processing OS was to transfer control from one job to another.
  - Problems

1. How does the monitor know about the nature of the job (e.g., Fortran versus Assembly) How does the monitor distinguish
   (a)one job from other job?
   (b) data from program?

  - Solution

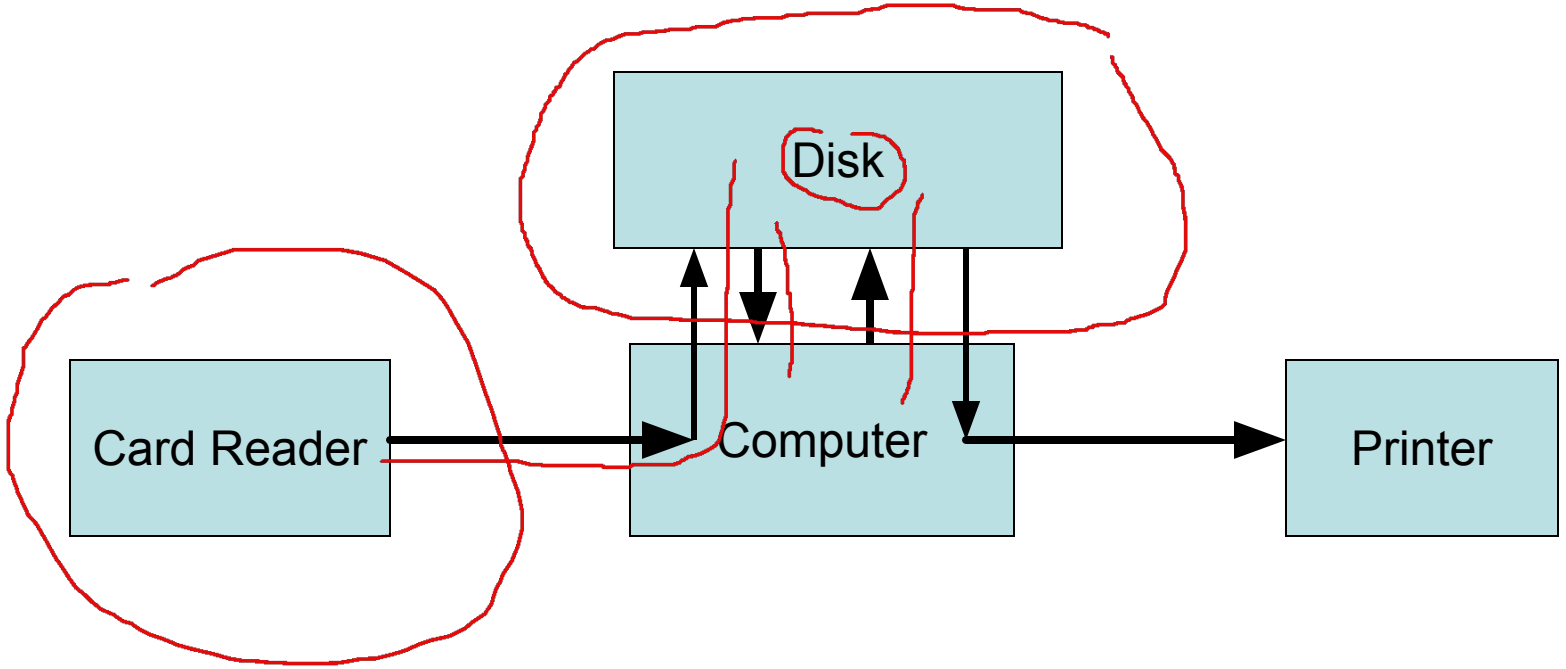
    Job Control Language

Turn around time in batch system



Throughput = (t6-t0)/n

Batch system with spooling



- Every executing program usually will have CPU Burst then I/O followed by CPU Burst and so on. The last burst is always CPU Burst and then program terminates
- c1 io1 c2 io2 c3--------cn

Problems of mainframe

- Early days Mainframes were inefficient as execution and I/O was done sequentially
- AS I/O devices are much slower than CPU execution , CPU utilization was poor

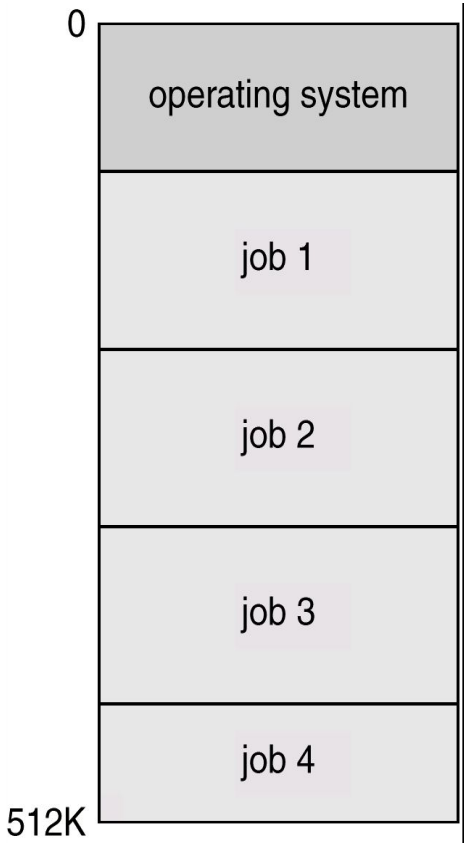
Solution

- Perform execution and I/O in concurrent manner
- Multiprogramming & Time sharing

Multiprogramming

- In this environment, multiple programs are kept in memory and one program is executed at a time. When a program terminates or goes for I/O resources, another program starts executing
- Multiprogramming improves CPU utilization
- Improves throughput

Memory Layout for multiprogramming



OS Features Needed for Multiprogramming

- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose amongst several jobs ready to run.
- Allocation of devices.

Time sharing

- A time unit is divided into small slice and each user can be allocated CPU in round robin manner
- A time shared OS allows many users to share the computer simultaneously
- Time shared OS uses multiprogramming and CPU scheduling to provide each user with a small portion of time shared computer
- Time sharing machine with Interactive I/O devices improves user response time
- Gives illusion that each user has his own machine

Requirements of time shared system

- Needs to keep track of time
- Requires Dynamic Resource allocation
  - To mange memory processes can be swapped out requiring backing store
- Dynamic resource allocation requires elaborate and complicated resource management and access control technique

Problems with early systems and change drivers

- High cost of resources
  - This necessitated concept of resource sharing and eventually led to the development of computer network
  - Technological development and IC fabrication has been main driver for development of low cost computers and variety of peripheral devices
  - Applications are main drivers for development of faster processor and computing system

Desktop systems

- Provided interaction
- Main concern is response time
- Since it serves single user, optimal resource utilization and protection are not a major concern
- Present day PCs support time sharing , multiprogramming and are network enabled

Parallel & Distributed System

- Parallel processing improves performance
- Approaches for parallel processing
  - Pipelining eg. Instruction pipelining
  - Multiplicity of components (Multiprocessor Systems)
    - Symmetric Multiprocessing
    - Asymmetric Multiprocessing
  - Multiplicity of Systems
    - Cluster
    - Grid

- Multiprocessor System Offers
  - Increased throughput
  - Economy
  - Increased Reliablity
    - Graceful Degradation
    - Fault tollerent
- Symmetric Multiprocessing :Each processor runs identical copy of OS
- Asymmetric multiprocessing: master slave relation exist.
- These are tightly coupled systems

Distributed Systems

- These systems depend on network for their functionality
  - Client Server Systems
  - Peer to peer systems
- Each processor has its own local memory
- Processors communicate with each other through communication links
- Loosely coupled Systems

Cluster System

- Cluster uses multiple CPUs to accomplish computational work
- Provide High availability
- Asymmetric Cluster (Hot Standby)
  - Hot standby machine monitors the active server and in case of failure of active server, hot standby becomes active server.
- Symmetric

Real Time system

- Hard Real time system
- Soft Real Time System