Quiz

CPU scheduling vs

Process CPU utilization (n, P, 1-p)

Online video lecture on 11/5 watch for quiz on wed. 11/ 7

Motivating concept for CPU scheduling

-maxi cpu utilization obtain with multiprog.

-CPU i/o burst cycle: process execution consists of cycle of CPU execution and I/o wait

-CPU burst distribution

**CPU Scheduler**

-Select from among process in mem that are ready to execute, allocate cpu to one of it

-CPU scheduling decision may take

\*switch from run to wait state

\*switch from running to ready state

\* switch from waiting to ready

\*terminates

-schedule under 1 and 4 is nonpreemptive

-all other scheduling is preemptive

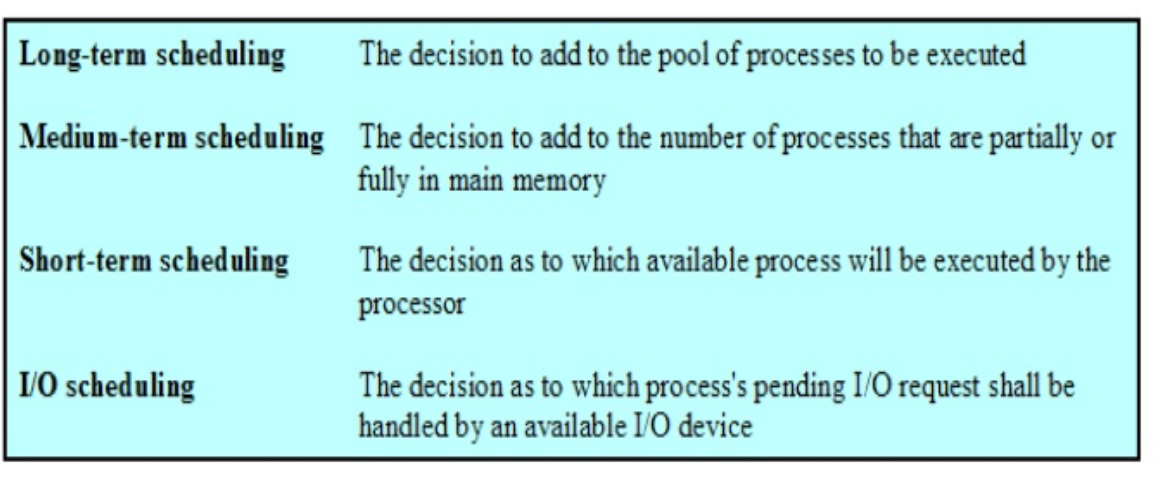
**Dispatcher**

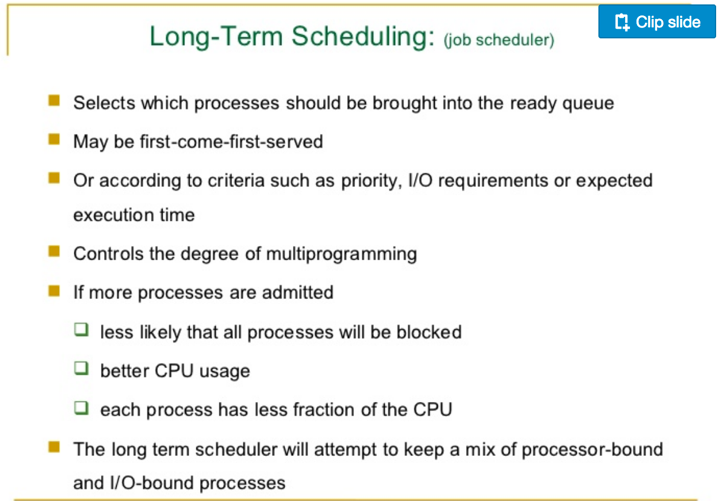
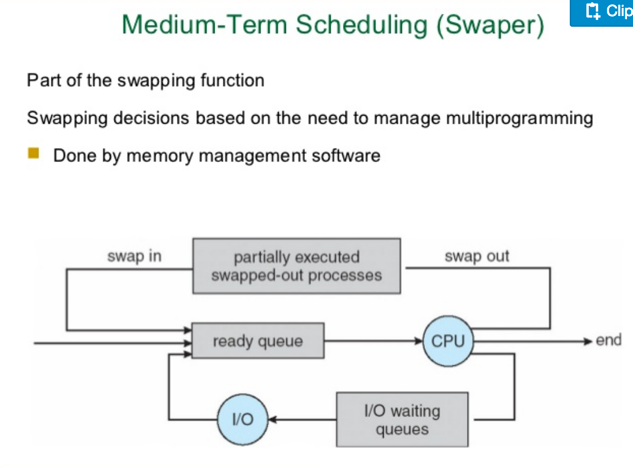
-dispatcher module give control of CPU to task( process / thread) selected by short term scheduler

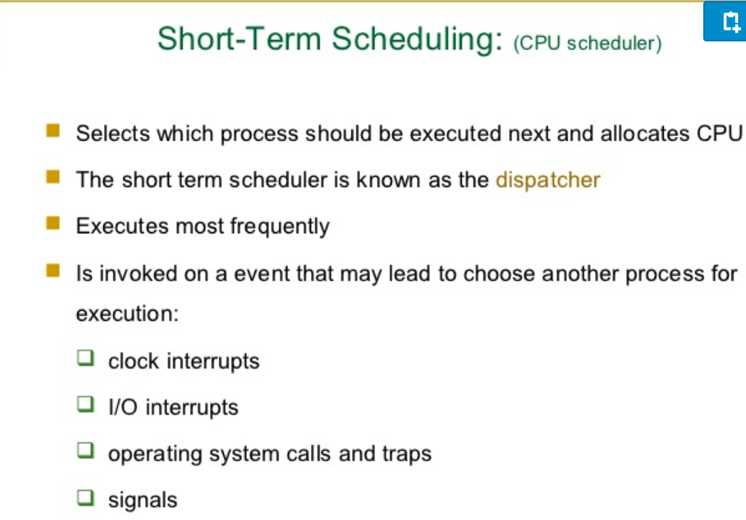
\*switch context

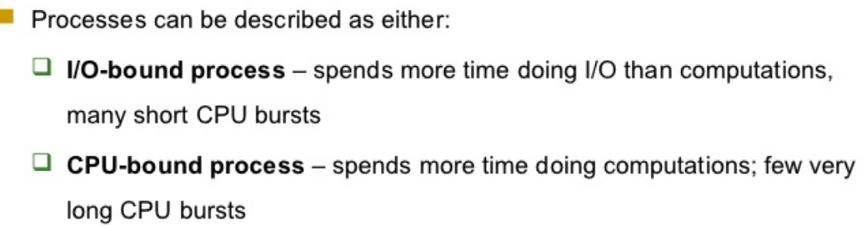
\*switch to user mode

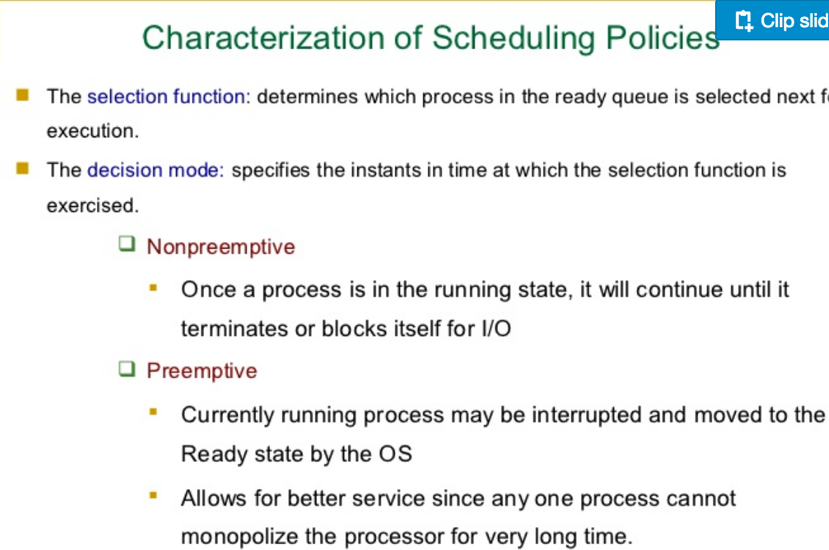
\*jumping to proper location in user program to restart program

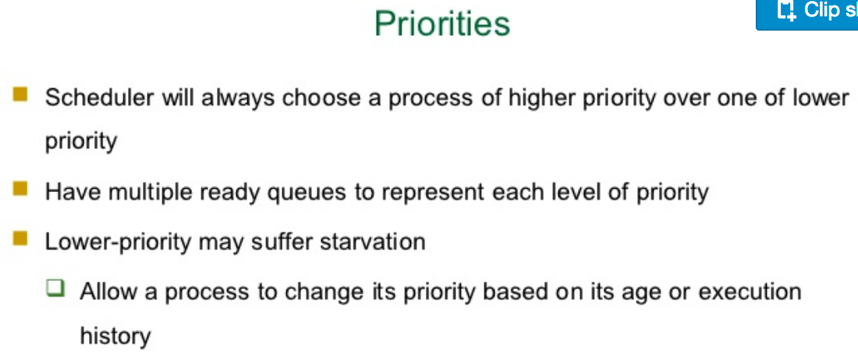


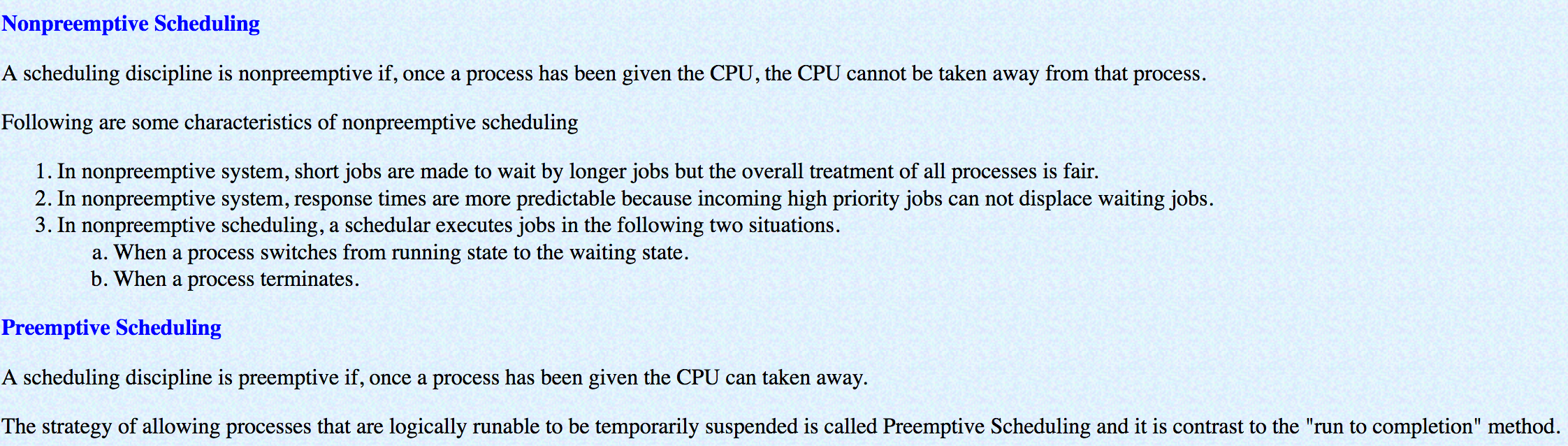


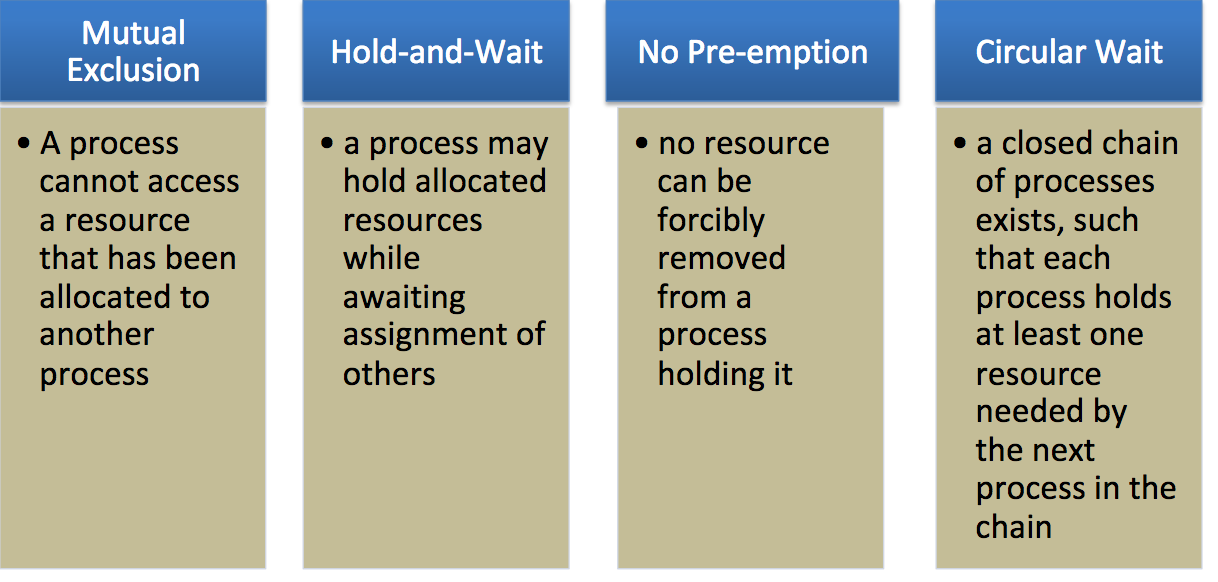










Condition for Deadlock

**Three general** approaches exist for dealing with deadlock:

-**Prevent Deadlock** : adopt a policy that eliminates one of the conditions

-**Avoid Deadlock** : make the appropriate dynamic choices based on the

current state of resource allocation

-**Detect Deadlock and Recover** : attempt to detect the presence of deadlock and take action to recover

**Non-preemptive vs Pre-emptive**

**-A preemptive** scheduling means that the

scheduler can take resources (e.g., CPU) away

from the process

-**A non-preemptive** scheduling means a process

occupies the resources until it voluntarily

relinquishes the resources

**Metrics to evaluate a scheduler**

-**Response time**: Time elapsed from the time of submission to the first

response

**Throughput**: # of tasks can be done per unit of time

**Wait time**: Time spent on waiting in the ready queue

**Turnaround time:** Tfinish – T start

**Processor Utilization:** Keep the CPU as busy as possible.

**Fairness**

**First Come First Serve (FCFS): ( non-preemptive)**

-Schedule tasks in the order they arrive

–Continue running them until they complete or give up

the processor

-Easy to implement; very small overhead due to

scheduling

-Average waiting time can be large if small jobs

wait behind long ones (high turnaround time)

–You have a basket, but wait behind one with a cart

**Shortest-Process-First (SPF) Scheduling**

-Scheduler selects process with smallest time to

finish

-Advantages: low average wait time

-Disadvantages:

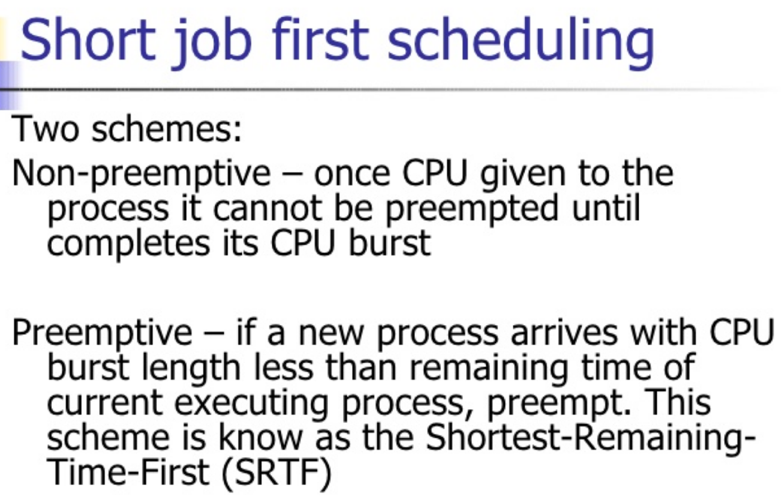
–Potentially large variance in wait times

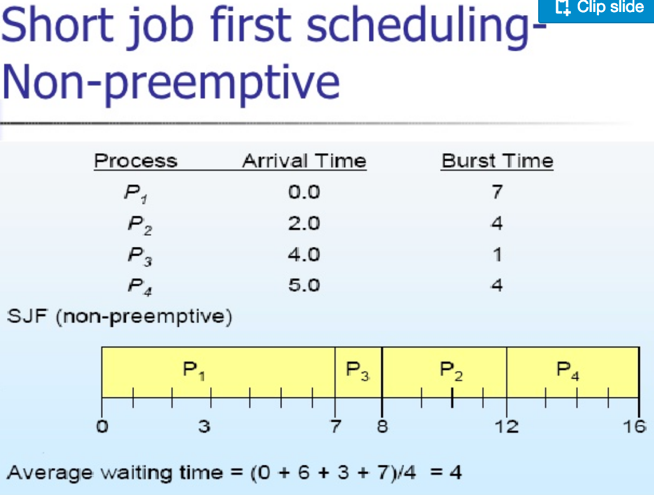
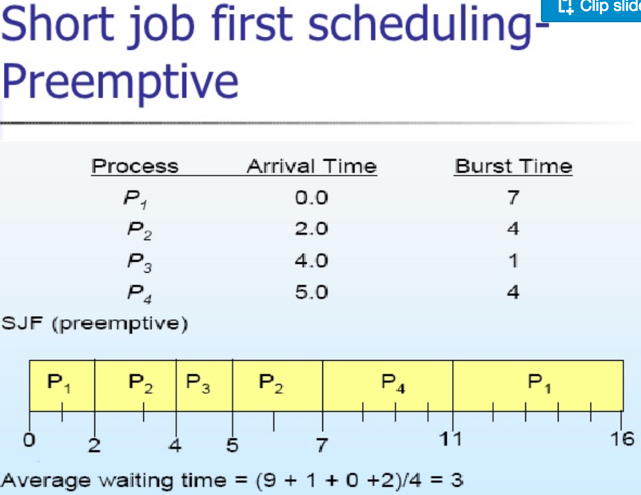
•Long task can be affected by short tasks once and again

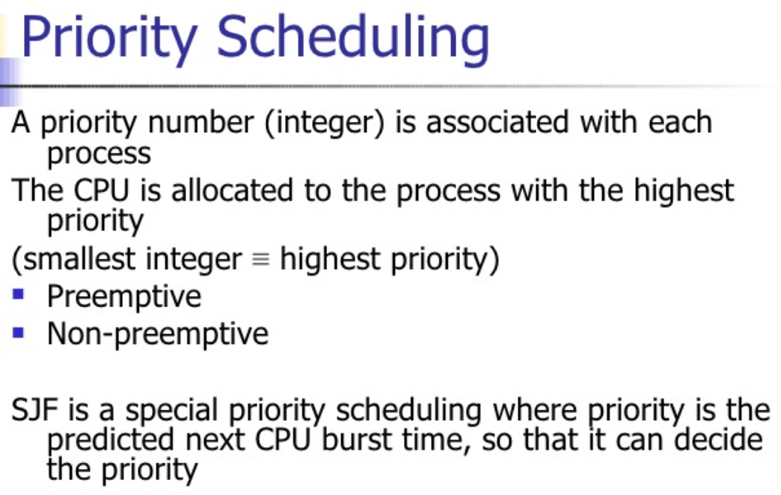
•Starvation

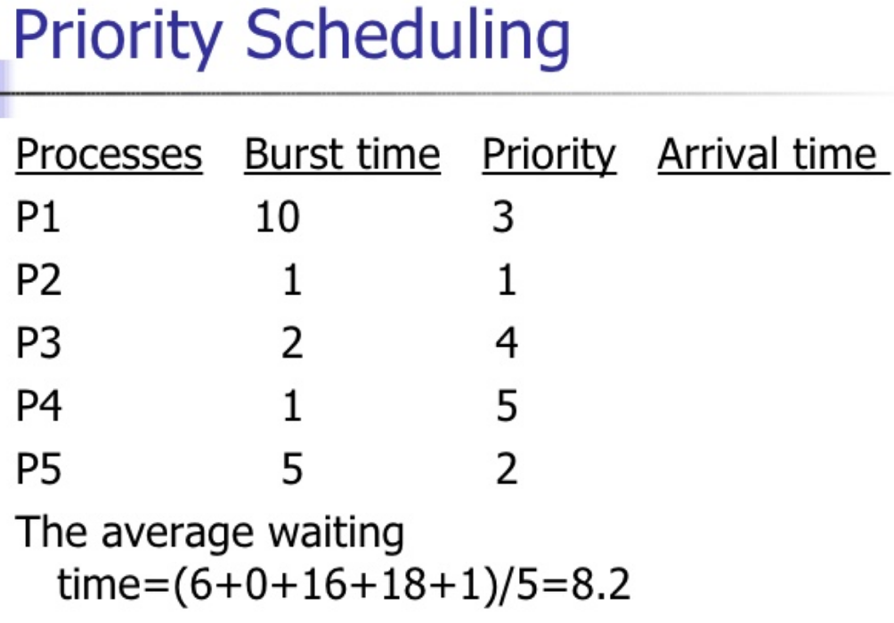
\*Relies on estimates of time-to-completion, which can

be inaccurate or **unrealistic**

****



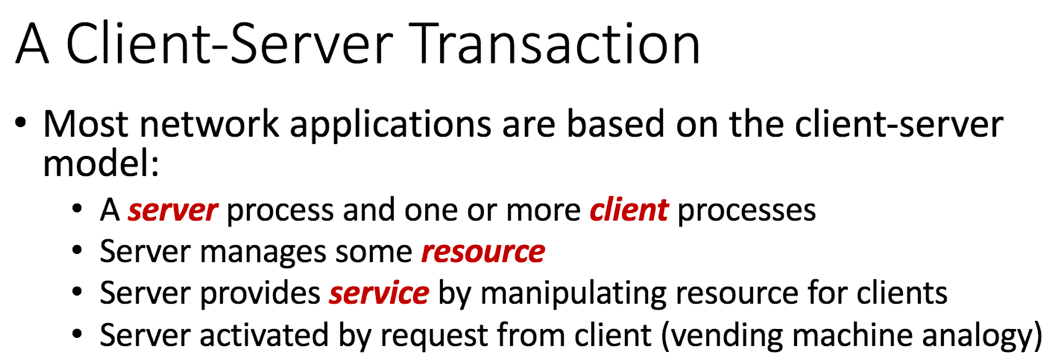


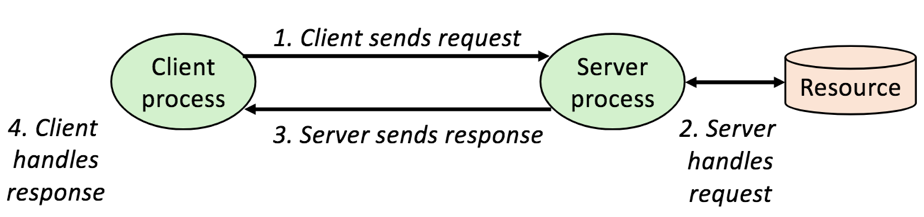


-t(n) = actual length

-

Network Programming (chapter 11)

****



**What Does an internet Protocol Do ?**

-**Provides a naming scheme**: An internet protocol defines a uniform format for **host addresses**

-**Provides a delivery mechanism:**

-An internet protocol defines a standard transfer unit **(packet)**

-Packet consists of **header and payload**

-Header: contains info such as packet size, source and destination addresses

-Payload: contains data bits sent from source host

**A Programmer’s View of the Internet**

1. Hosts are mapped to a set of 32 -**bit IP addresses**

-128.2.203.179

2. The set of IP addresses is mapped to a set of identifiers called

**Internet domain** names

-128.2.203.179 is mapped to www.cs.cmu.edu

3. A process on one Internet host can communicate with a process on

another Internet host over **a connection**

