# Class Note 10/21

Exam 2: Chapter 6-9 and Assignment 2

1. Using Dynamic Linking llib over static linking:

Advantages:

* take up space share between exe to update easily,
* security fix

Disadvantage:

* dependence of earlier library that are subject to change
* during update, updating process may fail

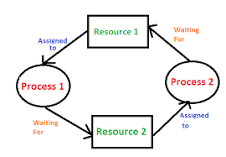
dll:

* Starts out as file
* When process using it, it is in

1. G Generate random number to check the value, ensure checking process does not start before generating by synchronization

Mutual exclusion: prevents simultaneous access to a shared resource.

1. In newer computer, race condition is a problem as something is deal with value created before\_\_\_
   1. Race Condition occurs when two or more threads can access shared data and they try to change it at the same time.
2. Process Deadlock: waiting for 2+ processes, be able to take the one thing waiting for and hold it, lock it, exclusions
   1. a set of **processes** are blocked because each **process** is holding a resource and waiting for another resource acquired by some other **process**



* 1. Sharing
  2. Waiting
  3. Holding
  4. Lock/Excluding

CG

Process that gen process that checks by \_\_\_

**Non-Preempt**

Arrive->Run

1. FCFS

1->2->3->4

Wait time on process 4: run at time 6 something else 4

TA time on process 4: finish at 8x8, run at 4, 8-4 = 4?

1. SJF (start at process 1 then find smallest of the next)

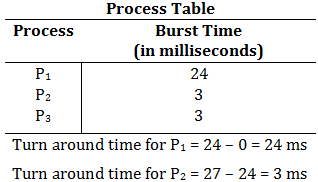
1->3->2……

1. Priorities (higher number of priority first)

Wait Time= (Start Time – Arrival Time) = (Turnaround time – Burst Time/Running time):

* Priority
* FCFS (3+2+1+3+2+1)

Turnaround Time P6 (Previous Process Finish Time – Previous Process Arrival Time):



waitingTime = startTime - arrivalTime

turnaroundTime = burstTime + waitingTime = finishTime- arrivalTime

Round Robin, no wait, 1 process to another, if have time, move to the new arrival time

* If Process 4 is AT 5sec
  + 1->1->2->3->1->4
  + Go back to process 1 if there is no new sequential arrival time for new process
  + Context Switches: when CPU switches from one process/thread to another
    - Used to save states of preempted processes

To run multiple instances that gen process in consumer process/cluster, they communicate by

* Not piping or share file
* Multiple Thread
  + Kernel
  + User
    - Cannot use for no shared memory
    - Kernel Level Thread Schedules Kernel level thread
    - User Level Thread schedule by cruntime etc…
  + Affect of choosing K/U Level
    - Performance (user faster: 1 core)
      * Cannot use multiple cores as the program only knows about 1 core (not flexible)
  + 20M numbers (numbers.txt)
    - Find smallest value using 4 threads
      * Wall time (Not Compute Time)
        + Elapsed time (beginning to end of execution)
      * Handle Errors
        + Handle Data: read line by line

Best to use driver that opens file read itself or threat read 🡪Index (Do Seek)