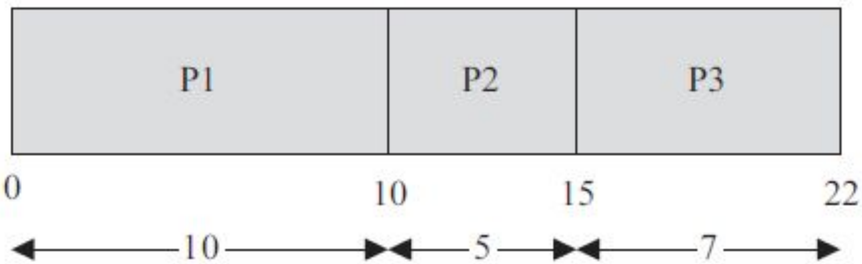
Embedded Systems

Dr. Sajesh Kumar U.

Non-Preemptive Scheduling

- Currently running process is allowed to run until it terminates or enters into wait state waiting for an I/O data or system resource.
- First-Come-First-Served (FCFS)/ FIFO Scheduling
- Last-Come-First Served (LCFS)/LIFO Scheduling
- Shortest Job First (SJF) Scheduling
- Priority Based Scheduling

First-Come-First-Served (FCFS)/ FIFO Scheduling



- **All 3 processes occurring at the same time in the order P1, P2, P3. P1 is executed first**

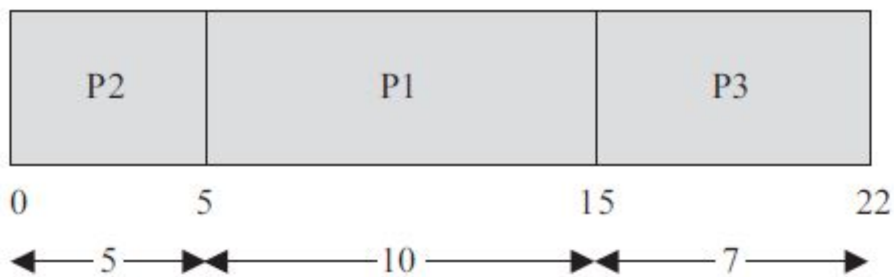
Waiting Time for P1 = 0 ms (P1 starts executing first)
Waiting Time for P2 = 10 ms (P2 starts executing after completing P1)
Waiting Time for P3 = 15 ms (P3 starts executing after completing P1 and P2)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P1+P2+P3)) / 3
= (0+10+15)/3 = 25/3
= 8.33 milliseconds

Turn Around Time (TAT) for P1 = 10 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P2 = 15 ms (-Do-)
Turn Around Time (TAT) for P3 = 22 ms (-Do-)
Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes
= (Turn Around Time for (P1+P2+P3)) / 3
= (10+15+22)/3 = 47/3
= 15.66 milliseconds

Average Turn Around Time (TAT) is the sum of average waiting time and average execution time.
Average Execution Time = (Execution time for all processes)/No. of processes
= (Execution time for (P1+P2+P3))/3
= (10+5+7)/3 = 22/3
= 7.33

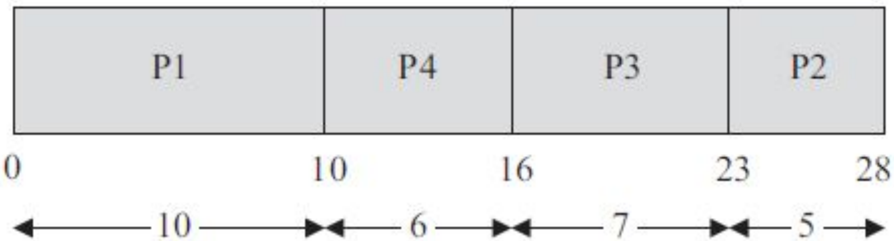
Average Turn Around Time = Average waiting time + Average execution time
= 8.33 + 7.33

First-Come-First-Served (FCFS)/ FIFO Scheduling



Waiting Time for P2 = 0 ms (P2 starts executing first)
Waiting Time for P1 = 5 ms (P1 starts executing after completing P2)
Waiting Time for P3 = 15 ms (P3 starts executing after completing P2 and P1)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P2+P1+P3)) / 3
= (0+5+15)/3 = 20/3
= 6.66 milliseconds
Turn Around Time (TAT) for P2 = 5 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P1 = 15 ms (-Do-)
Turn Around Time (TAT) for P3 = 22 ms (-Do-)
Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes
= (Turn Around Time for (P2+P1+P3)) / 3
= (5+15+22)/3 = 42/3
= 14 milliseconds

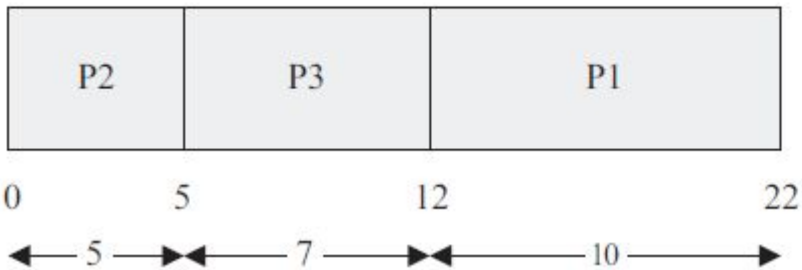
Last-Come-First Served (LCFS)/LIFO Scheduling



- P1 comes first, P2, P3 enters little later, P4 enters after 5ms

Waiting Time for P1 = 0 ms (P1 starts executing first)
Waiting Time for P4 = 5 ms (P4 starts executing after completing P1. But P4 arrived after 5 ms of execution of P1. Hence its waiting time = Execution start time – Arrival Time = 10 – 5 = 5)
Waiting Time for P3 = 16 ms (P3 starts executing after completing P1 and P4)
Waiting Time for P2 = 23 ms (P2 starts executing after completing P1, P4 and P3)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P1+P4+P3+P2)) / 4
= (0 + 5 + 16 + 23)/4 = 44/4
= 11 milliseconds
Turn Around Time (TAT) for P1 = 10 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P4 = 11 ms (Time spent in Ready Queue + Execution Time = (Execution Start Time – Arrival Time) + Estimated Execution Time = (10 – 5) + 6 = 5 + 6)
Turn Around Time (TAT) for P3 = 23 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P2 = 28 ms (Time spent in Ready Queue + Execution Time)
Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes
= (Turn Around Time for (P1+P4+P3+P2)) / 4
= (10+11+23+28)/4 = 72/4
= 18 milliseconds

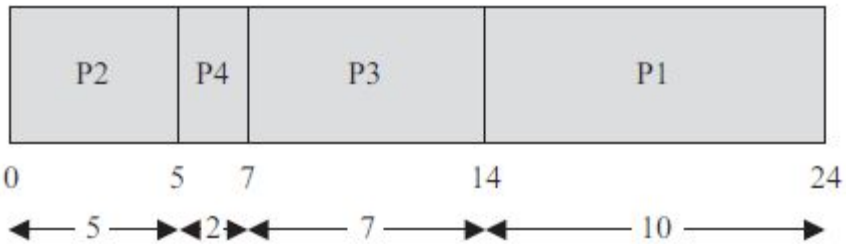
Shortest Job First (SJF) Scheduling



- P1, P2, P3 enters together. P2 with shortest time is executed first

Waiting Time for P2 = 0 ms (P2 starts executing first)
Waiting Time for P3 = 5 ms (P3 starts executing after completing P2)
Waiting Time for P1 = 12 ms (P1 starts executing after completing P2 and P3)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P2+P3+P1)) / 3
= (0+5+12)/3 = 17/3
= 5.66 milliseconds
Turn Around Time (TAT) for P2 = 5 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P3 = 12 ms (-Do-)
Turn Around Time (TAT) for P1 = 22 ms (-Do-)
Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes
= (Turn Around Time for (P2+P3+P1)) / 3
= (5+12+22)/3 = 39/3
= 13 milliseconds

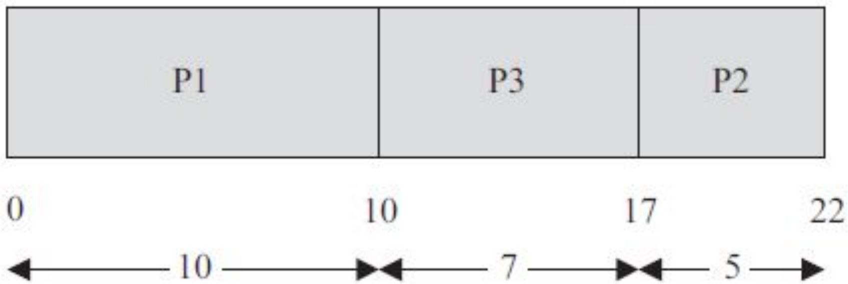
Shortest Job First (SJF) Scheduling



- P1, P2, P3 enters together, P4 comes after 2ms

Waiting time for P2 = 0 ms (P2 starts executing first)
Waiting time for P4 = 3 ms (P4 starts executing after completing P2. But P4 arrived after 2 ms of execution of P2. Hence its waiting time = Execution start time – Arrival Time = 5 – 2 = 3)
Waiting time for P3 = 7 ms (P3 starts executing after completing P2 and P4)
Waiting time for P1 = 14 ms (P1 starts executing after completing P2, P4 and P3)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P2+P4+P3+P1)) / 4
= (0 + 3 + 7 + 14)/4 = 24/4
= 6 milliseconds
Turn Around Time (TAT) for P2 = 5 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P4 = 5 ms (Time spent in Ready Queue + Execution Time = (Execution Start Time – Arrival Time) + Estimated Execution Time = (5 – 2) + 2 = 3 + 2)
Turn Around Time (TAT) for P3 = 14 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P1 = 24 ms (Time spent in Ready Queue + Execution Time)
Average Turn Around Time = (Turn Around Time for all Processes) / No. of Processes
= (Turn Around Time for (P2+P4+P3+P1)) / 4
= (5+5+14+24)/4 = 48/4
= 12 milliseconds

Priority Based Scheduling

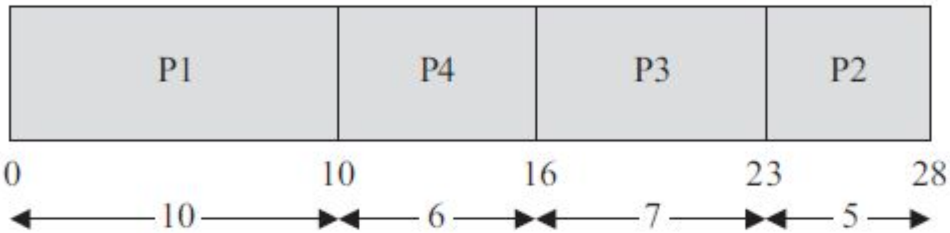


P1, P2,P3 with priority 0,3,2.
P1 with higher priority

Waiting time for P1 = 0 ms (P1 starts executing first)
Waiting time for P3 = 10 ms (P3 starts executing after completing P1)
Waiting time for P2 = 17 ms (P2 starts executing after completing P1 and P3)
Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P1+P3+P2)) / 3
= (0+10+17)/3 = 27/3
= 9 milliseconds
Turn Around Time (TAT) for P1 = 10 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P3 = 17 ms (-Do-)
Turn Around Time (TAT) for P2 = 22 ms (-Do-)
Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes

= (Turn Around Time for (P1+P3+P2)) / 3
= (10+17+22)/3 = 49/3
= 16.33 milliseconds

Priority Based Scheduling



- P1, P2,P3 with priority 0,3,2. P1 with higher priority. P4 with priority 1 enters after 5ms

Waiting time for P1 = 0 ms (P1 starts executing first)
Waiting time for P4 = 5 ms (P4 starts executing after completing P1. But P4 arrived after 5 ms of execution of P1. Hence its waiting time = Execution start time – Arrival Time = 10 – 5 = 5)
Waiting time for P3 = 16 ms (P3 starts executing after completing P1 and P4)
Waiting time for P2 = 23 ms (P2 starts executing after completing P1, P4 and P3)

Average waiting time = (Waiting time for all processes) / No. of Processes
= (Waiting time for (P1+P4+P3+P2)) / 4
= (0 + 5 + 16 + 23)/4 = 44/4
= 11 milliseconds

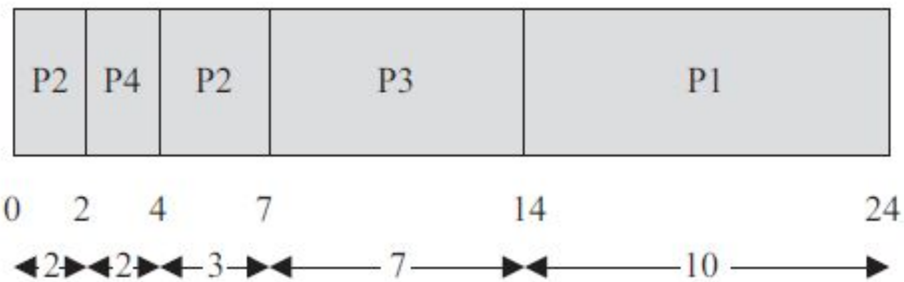
Turn Around Time (TAT) for P1 = 10 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P4 = 11 ms (Time spent in Ready Queue + Execution Time = (Execution Start Time – Arrival Time) + Estimated Execution Time = (10 – 5) + 6 = 5 + 6)
Turn Around Time (TAT) for P3 = 23 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P2 = 28 ms (Time spent in Ready Queue + Execution Time)

Average Turn Around Time = (Turn Around Time for all processes) / No. of Processes
= (Turn Around Time for (P2 + P4 + P3 + P1)) / 4
= (10 + 11 + 23 + 28)/4 = 72/4
= 18 milliseconds

Pre-emptive Scheduling

- In Pre-emptive each process in ready queue gets a chance to execute. When and how often a process gets chance depends on scheduling algorithm
- Preemptive SJF Scheduling/Shortest remaining time
- Round Robin Scheduling
- Priority based Scheduling

Preemptive SJF Scheduling/Shortest remaining time



- P1,P2,P3 with execution time 10,5,7 enters first. P4 with execution time 2ms, enters after 2ms

Waiting time for P2 = 0 ms + (4 – 2) ms = 2 ms (P2 starts executing first and is interrupted by P4 and has to wait till the completion of P4 to get the next CPU slot)

Waiting time for P4 = 0 ms (P4 starts executing by preempting P2 since the execution time for completion of P4 (2 ms) is less than that of the Remaining time for execution completion of P2 (Here it is 3 ms))

Waiting time for P3 = 7 ms (P3 starts executing after completing P4 and P2)

Waiting time for P1 = 14 ms (P1 starts executing after completing P4, P2 and P3)

Average waiting time = (Waiting time for all the processes) / No. of Processes
= (Waiting time for (P4+P2+P3+P1)) / 4
= (0 + 2 + 7 + 14)/4 = 23/4
= 5.75 milliseconds

Turn Around Time (TAT) for P2 = 7 ms (Time spent in Ready Queue + Execution Time)

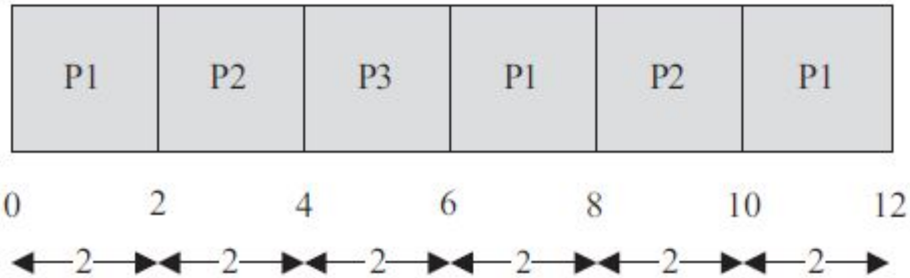
Turn Around Time (TAT) for P4 = 2 ms (Time spent in Ready Queue + Execution Time = (Execution Start Time – Arrival Time) + Estimated Execution Time = (2 – 2) + 2)

Turn Around Time (TAT) for P3 = 14 ms (Time spent in Ready Queue + Execution Time)

Turn Around Time (TAT) for P1 = 24 ms (Time spent in Ready Queue + Execution Time)

Average Turn Around Time = (Turn Around Time for all the processes) / No. of Processes
= (Turn Around Time for (P2+P4+P3+P1)) / 4
= (7+2+14+24)/4 = 47/4
= 11.75 milliseconds

Round Robin Scheduling



- P1, P2, P3 with execution times 6ms,4ms,2ms. Round robin time slice is 2ms

Waiting time for P1 = 0 + (6 – 2) + (10 – 8) = 0 + 4 + 2 = 6 ms
(P1 starts executing first and waits for two time slices to get execution back and again 1 time slice for getting CPU time)

Waiting time for P2 = (2 – 0) + (8 – 4) = 2 + 4 = 6 ms
(P2 starts executing after P1 executes for 1 time slice and waits for two time slices to get the CPU time)

Waiting time for P3 = (4 – 0) = 4 ms
(P3 starts executing after completing the first time slices for P1 and P2 and completes its execution in a single time slice)

Average waiting time = (Waiting time for all the processes) / No. of Processes
= (Waiting time for (P1 + P2 + P3)) / 3
= (6 + 6 + 4)/3 = 16/3
= 5.33 milliseconds

Turn Around Time (TAT) for P1 = 12 ms (Time spent in Ready Queue + Execution Time)
Turn Around Time (TAT) for P2 = 10 ms (-Do-)
Turn Around Time (TAT) for P3 = 6 ms (-Do-)

Priority Based Scheduling

P1	P4	P1	P3	P2

0 5 11 16 23 28

←5→←6→←5→←7→←5→

P1,P2,P3 with estimated times 10,5,7ms and priorities 1,3,2. P4 with estimated time 6ms and priority 0 enters after 5ms

Waiting time for P1 = 0 + (11 – 5) = 0 + 6 = 6 ms
(P1 starts executing first and gets preempted by P4 after 5 ms and again gets the CPU time after completion of P4)

Waiting time for P4 = 0 ms
(P4 starts executing immediately on entering the 'Ready' queue, by preempting P1)

Waiting time for P3 = 16 ms (P3 starts executing after completing P1 and P4)

Waiting time for P2 = 23 ms (P2 starts executing after completing P1, P4 and P3)

Average waiting time = (Waiting time for all the processes) / No. of Processes
= (Waiting time for (P1+P4+P3+P2)) / 4
= (6 + 0 + 16 + 23)/4 = 45/4
= 11.25 milliseconds

Turn Around Time (TAT) for P1 = 16 ms (Time spent in Ready Queue + Execution Time)

Turn Around Time (TAT) for P4 = 6 ms
(Time spent in Ready Queue + Execution Time = (Execution Start Time – Arrival Time) + Estimated Execution Time = (5 – 5) + 6 = 0 + 6)

Turn Around Time (TAT) for P3 = 23 ms (Time spent in Ready Queue + Execution Time)

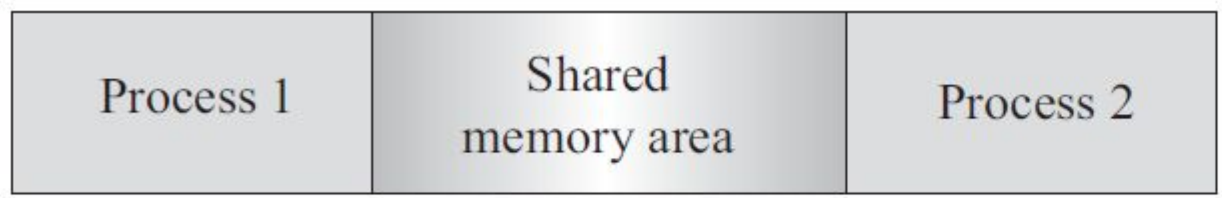
Turn Around Time (TAT) for P2 = 28 ms (Time spent in Ready Queue + Execution Time)

Average Turn Around Time = (Turn Around Time for all the processes) / No. of Processes
= (Turn Around Time for (P2 + P4 + P3 + P1)) / 4
= (16 + 6 + 23 + 28)/4 = 73/4
= 18.25 milliseconds

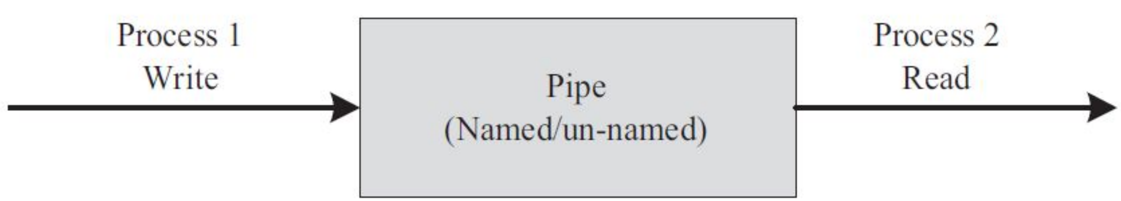
Task Communication

- Co-operating processes----Shared data between them
- Competing processes----compete for the system resources
- Co-operation through----- shared resources. Communication for synchronization
- Task communication/Inter process communication (IPC)
    - Shared Memory
    - Message Passing
    - Remote Procedure Call (RPC) and Sockets

Shared Memory

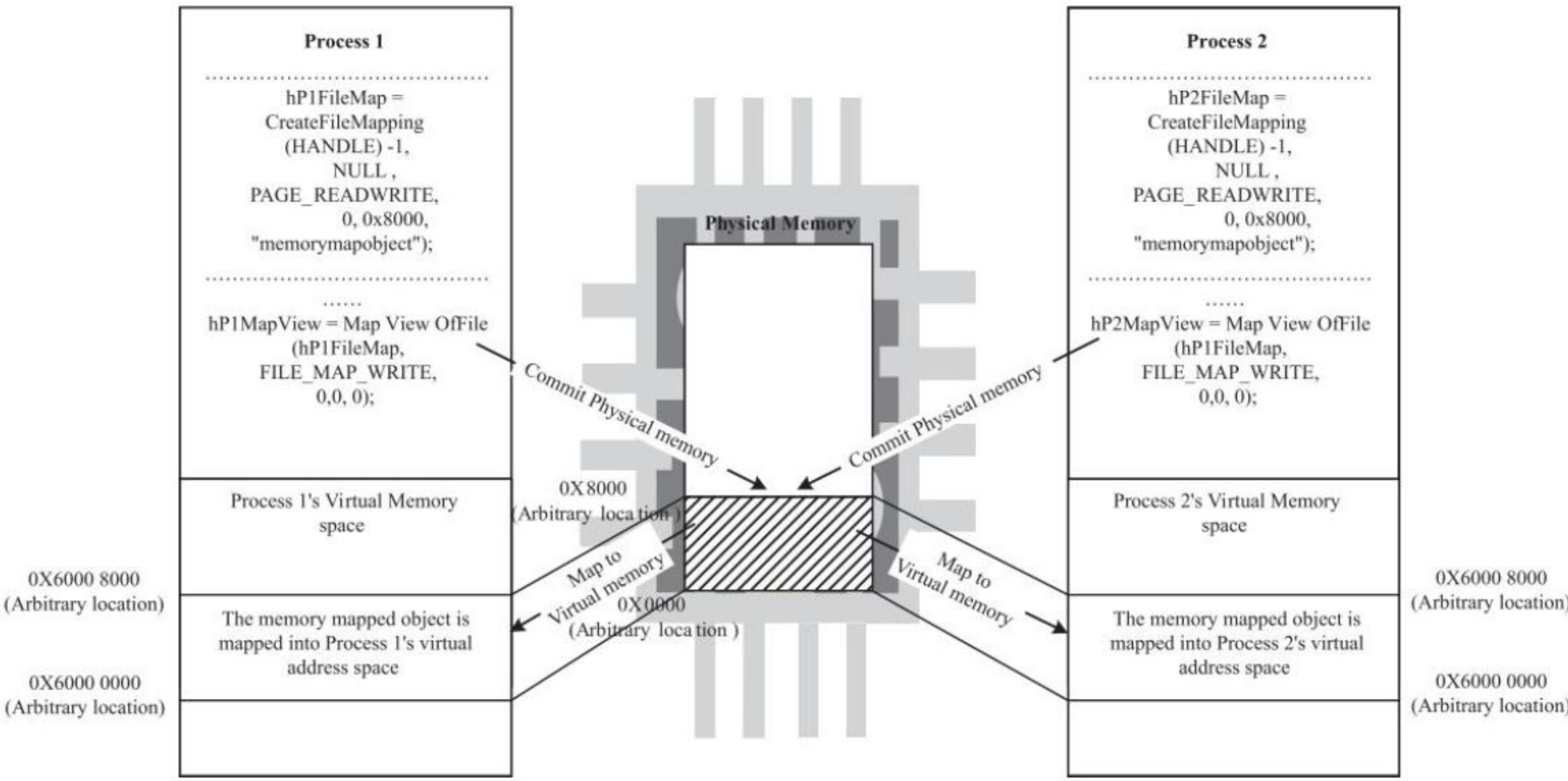


- **Pipes**-Client-server architecture
- Anonymous and Named pipes



Shared Memory

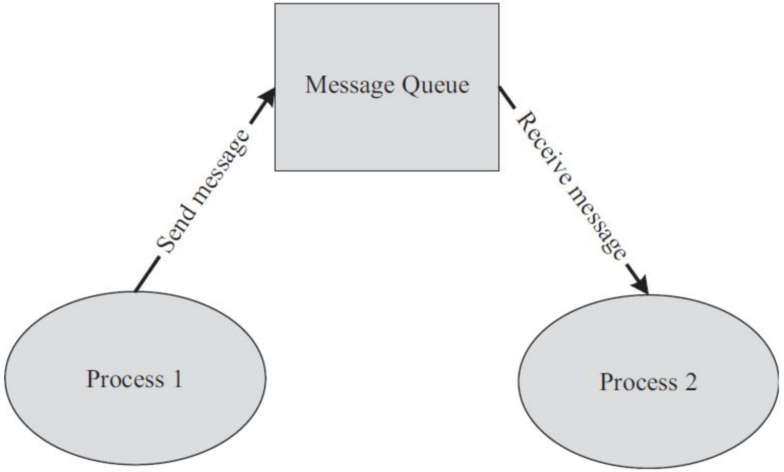
Memory Mapped Objects—Allocation of shared block of memory in virtual space



Message Passing

- Inter process/Thread communication
- Limited amount of data is passed
- Fast and free from synchronization
- **Message Queue**

Send and receive methods
Uses FIFO queue



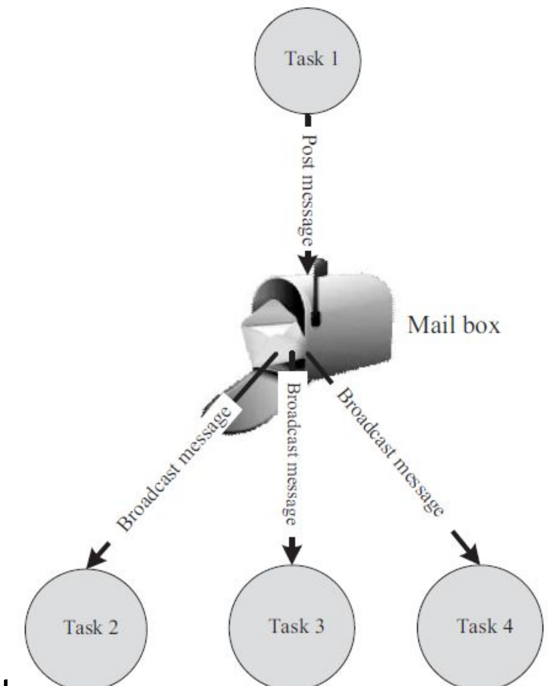
Message Passing

Mailbox

- creates a mail box for posting the messages
- Interested thread can subscribe the mailbox

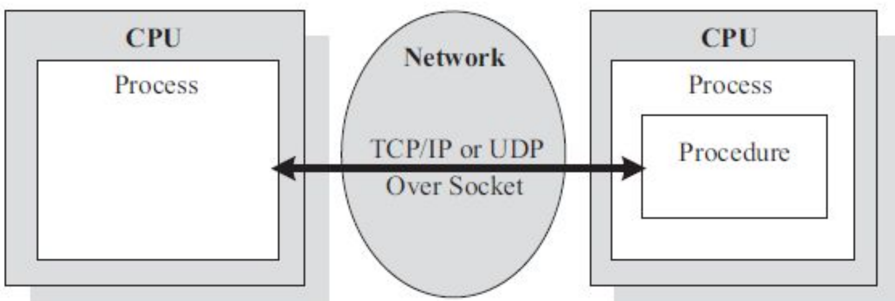
Signalling

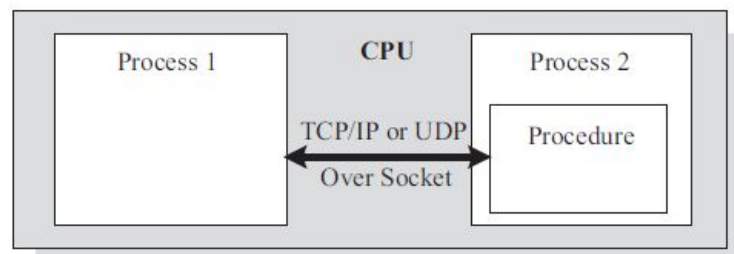
- An asynchronous signal is generated by a process indicating the scenario which other process/thread is waiting for



Remote Procedure Call (RPC) and Sockets

- Process call a procedure of another process running on the same CPU or another CPU
- Communication over heterogeneous network
- Synchronous or asynchronous





Socket is a logical end point in a 2 way communication
Stream sockets over TCP and datagram sockets over UDP

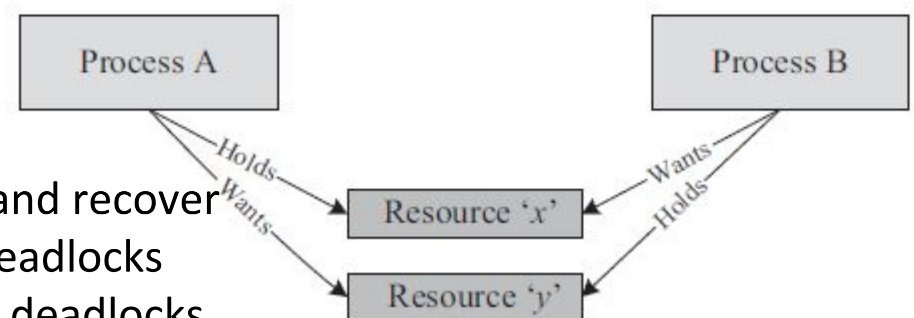
Task Synchronization

- Each process should be aware of the access of the shared resource while multiple processes are trying to access the shared resources
- Task synchronization is required
- Task synchronization issues may arise if not properly synchronized

Task Synchronization Issues

- Racing—multiple processes compete each other to access and manipulate the shared data concurrently
- Deadlock—None of the processes will be able to progress their execution

- Mutual exclusion
- Hold and Wait
- No resource Pre emption
- Circular wait

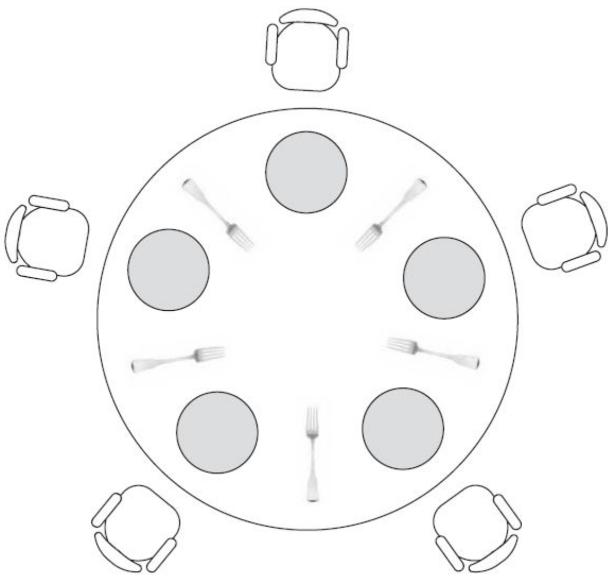


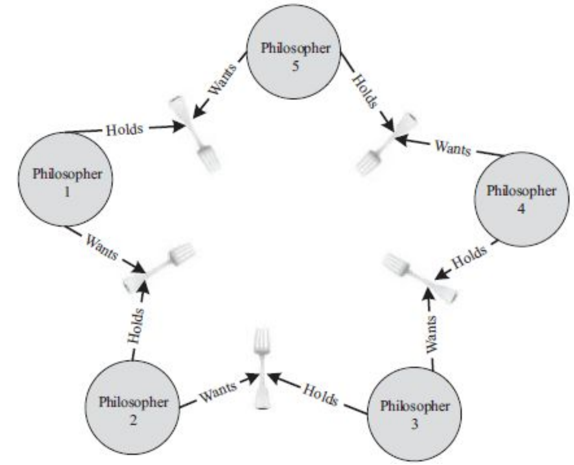
Detect and recover
Avoid deadlocks
Prevent deadlocks

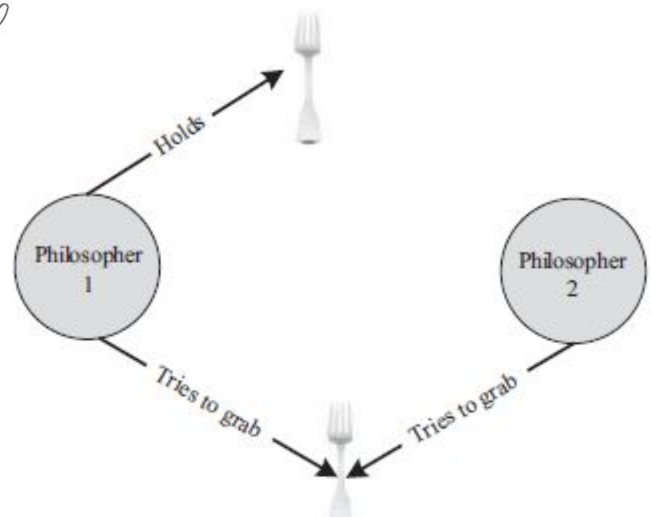
Live lock
Starvation

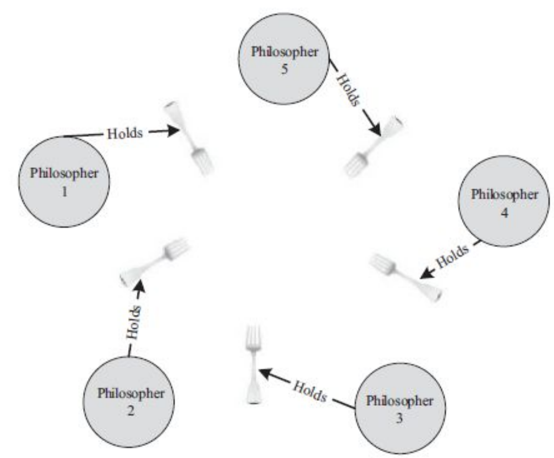
Task Synchronization Issues

- The Dining Philosophers Problem









Task Synchronization Issues

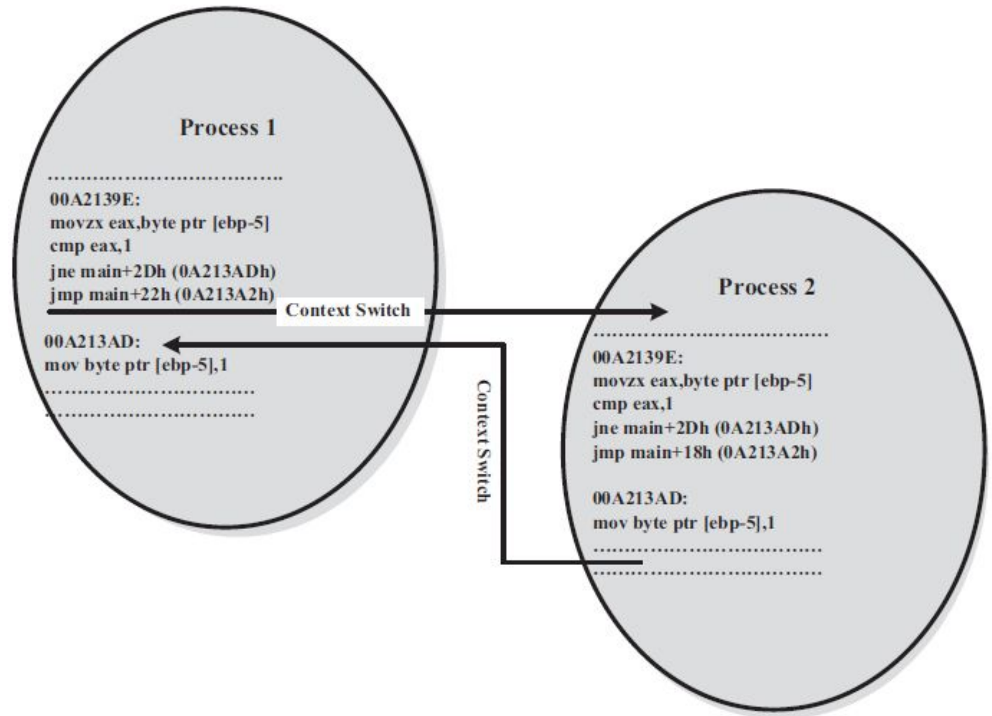
- **The Producer-Consumer/Bounded Buffer Problem**
- **Reader-Writers problem**
- **Priority Inversion**—blocking based synchronization and pre-emptive priority scheduling.
- Condition where higher priority task waiting for source held by low priority task and medium priority task always pre empts the low priority task.
- **Priority Inheritance---**Low priority task temporarily inherits the priority of high priority task
- **Priority ceiling—**Low priority tack which is accessing a shared resource will be elevated to higher priority ceiling of the resource.

Task synchronization Techniques

- Avoiding conflicts in resource access like racing, deadlock, starvation
- Ensure proper sequence of operation across processes
- Communicating between processes

Task synchronization Techniques

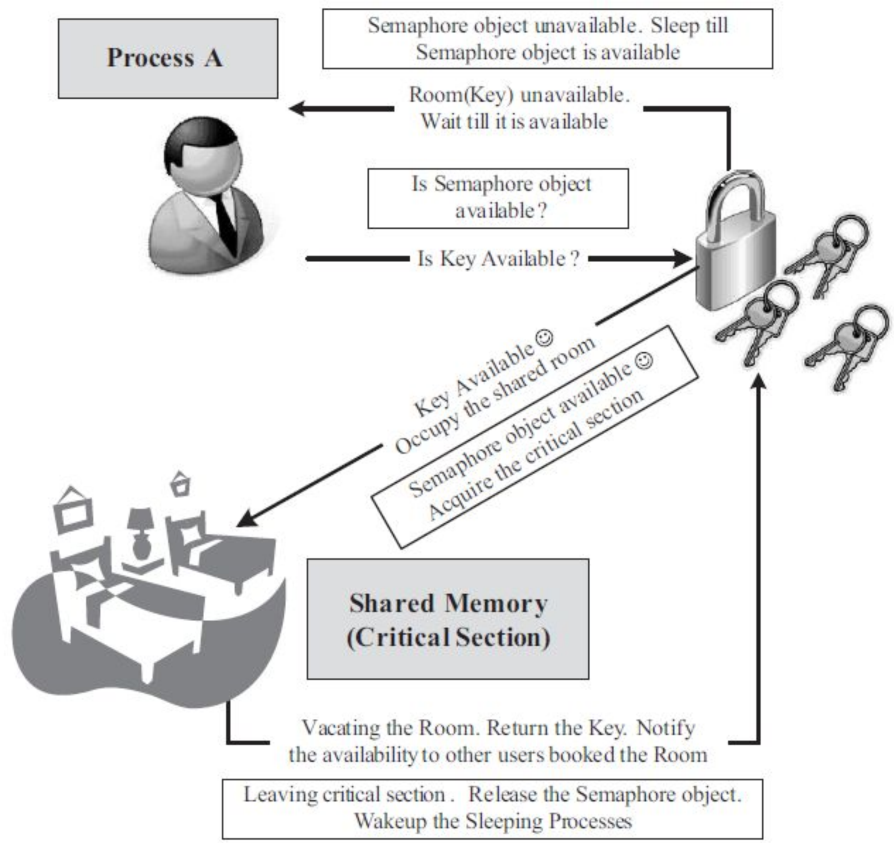
- Mutual exclusion through busy waiting---Using lock



Task synchronization Techniques

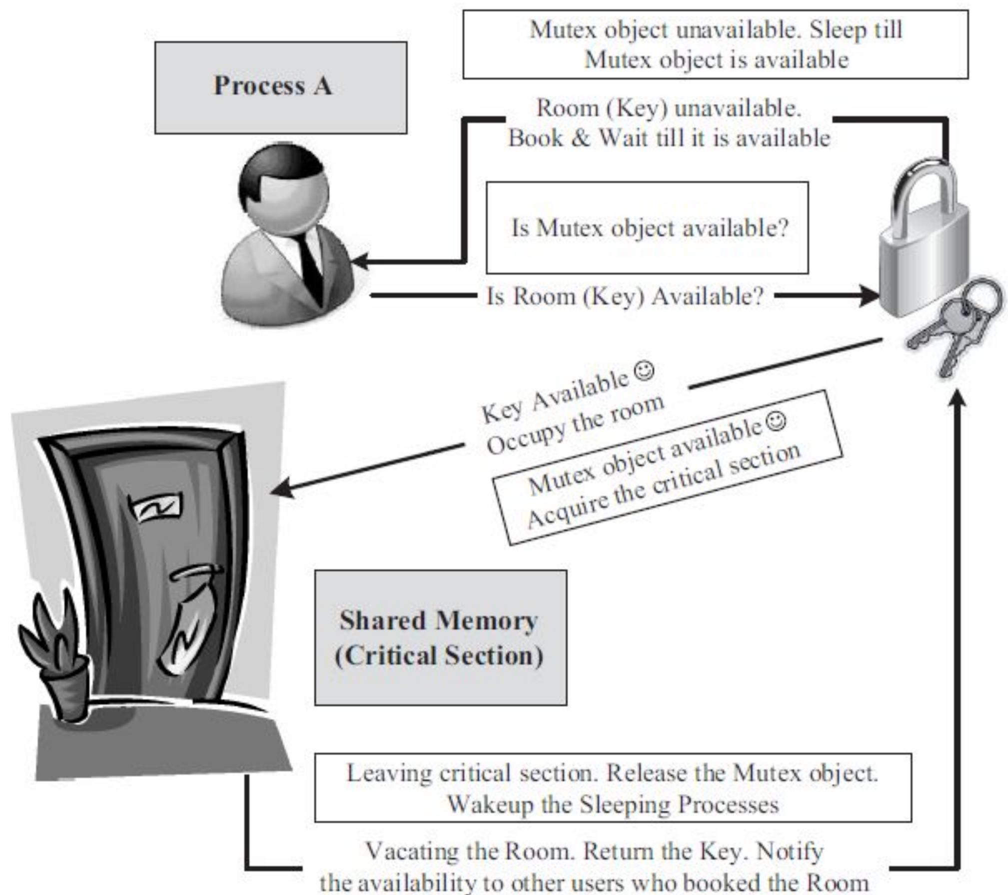
- Mutual exclusion through sleep and wakeup

- Semaphore
- Binary semaphore
- Counting semaphore



Task synchronization Techniques

- Binary semaphore (Mutex)

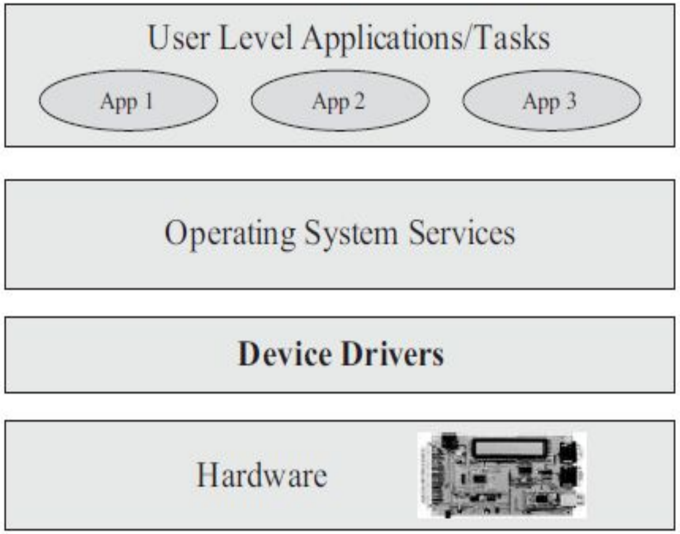


Device Driver

- Bridge between OS and Hardware
- Device drivers are OS dependent

Regardless of OS type, device drivers implement the following

- Device initialization and Interrupt configuration (Interrupt type, IRQ, ISR)
- Interrupt handling and processing
- Client interfacing(Inter process communication and synchronization)



Choosing an RTOS

Functional Requirements

- Processor support
- Memory requirements
- Real time capabilities
- Kernal and interrupt latency
- Inter process communication and Task Synchronization
- Modularization support
- Support for networking and communication
- Development language support

Choosing an RTOS

Non Functional Requirements

- Custom developed or off the shelf
- Cost
- Development and Debugging tools availability
- Ease of Use
- After Sales