DEPSTAR

OS Assignment (Submission Deadline Interval-30/10/2021)

Note- You need to solve Numerical only, theory question is for your reference.

Unit-01 Introduction to OS

|  |  |
| --- | --- |
| 1 | List basic services of an Operating System. |
| 2 | Explain the terms: (i) Shell (ii) Kernel |
| 3 | List Network Operating System Features and Functions. |
| 4 | In Linux, shared libraries perform many operations central to the operating system. What is the advantage of keeping this functionality out of the kernel? |
| 5 | List out the advantages of microkernel architecture. |
| 6 | Give the difference between multiprogramming and multiprocessing. |
| 7 | What is the difference between system call and system program? |

|  |  |
| --- | --- |
| 1 | What is an operating system? What are the functions of the operating system? |
| 2 | Explain architecture of UNIX environment. |
| 3 | Give the advantages of Distributed Operating System. |

|  |  |
| --- | --- |
| 1 | Give three different views of Operating System Application View: what services does it provide? System View: what problems does it solve?  Implementation View: how is it built? |
| 2 | Explain the objectives and functions of operating systems |
| 3 | What is operating system? Give the view of OS as a resource manager. |
| 4 | Explain Monolithic Architecture. |

Unit-02 Process Management & CPU Scheduling

|  |  |
| --- | --- |
| 1 | Discuss differences among short-term, medium term and long-term scheduling. |
| 2 | What are the differences between user-level and kernel level threads? Under what circumstances is one type better than other? |
| 3 | Define the disadvantages of User-Level Threads (ULTs) compared to Kernel Level Threads (KLTs). |

|  |  |
| --- | --- |
| 4 | List the reasons for process termination. |
| 5 | Draw and explain process state-transition diagram with suspended state. |
| 6 | Give comparison of non-preemptive and preemptive scheduling. |
| 7 | What are the benefits of multithreaded programming? Also explain about TCB (Thread  Control Block). |
| 8 | Define and differentiate process and thread. |
| 9 | What do you mean by scheduling? Discuss in brief types of scheduler. |

|  |  |
| --- | --- |
| 1 | What is a process? What are attributes of a process? |
| 2 | Assuming three states of a process (Running, Ready, Blocked), draw and explain the state transition diagram. |
| 3 | Write down the name of structure used by Operating System to maintain information about a process. Also write down any four-information related to specific process which is maintained in this structure. |
| 4 | Describe the actions taken by a kernel to context-switch between kernel level threads. |
| 5 | What are main difference between user-level threads and kernel-level threads? |
| 6 | what resource are used when a thread is created? How do they differ from those used when a process is created? |
| 7 | Differentiate between preemptive and no-preemptive CPU scheduling. Name how CPU scheduling algorithms for each type of scheduling. |
| 8 | Write down two advantages in having variant time-quantum sizes at different levels of a multilevel queuing system? |
| 9 | What is round robin scheduling? Explain using an example. Can it be useful for a single user system? Justify your answer. |
| 10 | What is PCB? Discuss its major fields |
| 11 | How does a parent process create a child process? How does a parent process create an  ordinary pipe (anonymous pipe) for communicating with child process? Write the steps |
| 12 | Explain Process Address Space with an example. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | **Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:** | | | | |
|  | Process | Burst  Time | Priority |  |
| P1 | 7 | 5 |
| P2 | 2 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | P3 | 3 | 3 | | | The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.  a) Draw four Gantt charts illustrating the execution of  these processes using FCFS, SJF (non-preemptive), a | | |
| P4 | 1 | 4 | | |
| P5 | 9 | 2 | | |
| non preemptive priority (a smaller priority number  implies a higher priority), and RR (**quantum = 1**) scheduling.   1. What is the turnaround time of each process for each of the scheduling algorithms in part (a) and average turnaround time? 2. What is the waiting time of each process for each of the scheduling algorithms in part (a) and average waiting time? 3. Which of the schedules in part a results in the minimal average waiting time (over all processes)? | | | | | | | | |
| 2 | **Consider the following set of processes, with the length of the CPU-burst time and the arrival time given in milliseconds:**   1. Draw four Gantt charts illustrating the execution   of these processes using FCFS, SJF (non- preemptive), SJF (preemptive), and RR (**quantum**  **= 2**) scheduling.   1. What is the turnaround time of each processfor each of the scheduling algorithms in part (a) and average turnaround time? 2. What is the waiting time of each process for   each of the scheduling algorithms in part (a) and average waiting time?   1. Which of the schedules in part a results in the minimal average waiting time (over all processes)? | | | | | | | | |
| 3 | Consider Five Processes P1 to P5 arrived at same time. They have estimated running time 10,2,6,8 and 4 seconds, respectively. Their Priorities are 3,2,5,4 and 1, respectively with 5 being  highest Priority.  Find the average turnaround time and average waiting time for Round Robin(**q=3**) and Priority Scheduling algorithm. | | | | | | | | |
| 4 | **Following table gives arrival time and expected run time of five processes. Ignore process switching overhead.** | | | | | | | | |
|  | Process | Arrival Time | | | Expected Run time | | | Find average turnaround time for  following scheduling algorithm.   1. Round robin (**quantum = 1 sec**) 2. Shortest |
| A | 1 | | | 8 | | |
| B | 4 | | | 1 | | |
| C | 2 | | | 2 | | |
| D | 5 | | | 1 | | |
| E | 6 | | | 5 | | |
| 5 | **Differentiate between preemptive and non-preemptive scheduling.** | | | | | | | | |
|  | Process | CPU Time | | Arrival Time | | | 1. Solve following by SJF preemptive and non- preemptive. 2. Draw Gantt Chart, Average Waiting Time and Average Turnaround Time. c) Which one is better as per average turnaround time? | |
| P1 | 6 | | 0 | | |
| P2 | 4 | | 1 | | |
| P3 | 5 | | 3 | | |
| P4 | 3 | | 5 | | |

|  |  |  |
| --- | --- | --- |
| Process | CPU Time | Arrival Time |
| P1 | 7 | 5 |
| P2 | 2 | 1 |
| P3 | 3 | 3 |
| P4 | 1 | 4 |
| P5 | 9 | 2 |

|  |  |
| --- | --- |
| 6 | Consider the following set of processes in order P1, P2, P3, P4 and P5 with the length of the CPU burst time given in milliseconds. Calculate average waiting time and turn-around time using Shortest Job First (Pre-emptive) and Round Robin (quantum=3ms) scheduling algorithm.  Draw Gantt chart for both the algorithms.  **Process Arrival Time Burst Time**  P1 0 9  P2 1 7  P3 2 6  P4 3 7  P5 4 3 |
| 7 | Given the following information:    a) Draw a Gantt chart (time-line) for each of the following scheduling algorithms.   * First-come first-served * Shortest job first * Round robin (using a time quantum of 4)   b) For the round robin algorithm above, compute the waiting time and turnaround time for each process  (not the average). |

Unit-03 Process Synchronization

|  |  |
| --- | --- |
| 1 | What is monitor? How is it used to access critical section? |
| 2 | What do you mean by process synchronization? Why is it required? |
| 3 | What are the advantages of inter-process communication? |
| 4 | Define Race condition. |
| 5 | How to achieve mutual exclusion using strict alternation? |
| 6 | Differentiate semaphore and mutex. |
| 7 | Differentiate semaphore and monitor. |

|  |  |
| --- | --- |
| 1 | What is the meaning of the term busy waiting? What other kinds of waiting are there in an  operating system? Can busy waiting be avoided altogether? Explain your answer. |
| 2 | What is critical section problem? What are the requirements that a solution to critical  section problem must satisfy? |
| 3 | Define the following: |

|  |  |
| --- | --- |
|  | i) Wait & Signal Operation ii) Semaphores iii) Race Condition. |
| 4 | Explain producer-consumer problem and solve it using semaphore. Write pseudo code for the  same. |
| 5 | Explain IPC Problem – Readers & Writers Problem. |
| 6 | What is Mutex? Write a pseudo code to achieve mutual exclusion using mutex. |
| 7 | Define mutual exclusion. How mutual exclusion can be achieved? Explain. |
| 8 | What is race condition? Explain producer – consumer problem with fatal race condition. |
| 9 | Write a short note on Critical Section. |
| 10 | What is Mutual exclusion? Explain Peterson’s solution for mutual exclusion problem. |

|  |  |
| --- | --- |
| 1 | How to solve producer-consumer problem with semaphore. Explain in detail |
| 2 | Explain working flow of dinning philosopher problem using mutex. What is the drawback of  this solution? |
| 3 | Explain working flow of dinning philosopher problem using semaphore. |
| 4 | What is Semaphore? Give the implementation of Readers-Writers Problem using Semaphore. |

Unit-04 Deadlock

|  |  |
| --- | --- |
| 1 | Is it possible to have a deadlock involving only one single process? Explain your answer. |
| 2 | Consider a system consisting of 4 resources of the same type that are shared by 3 processes,  each of which needs at most 2 resources. Show that the system is deadlock free. |
| 3 | Can a system detect that some of its processes are starving? If you answer “yes,” explain how  it can. If you answer “no,” explain how the system can deal with the starvation problem. |
| 4 | Differentiate between deadlock and starvation. |
| 5 | If each resource type has several instances, then a cycle does not necessarily imply that a  deadlock has occurred. Explain with resource allocation graph. |

|  |  |
| --- | --- |
| 1 | Given the following resource allocation diagram,    a. If another instance of resource r1 is made available, is the deadlock resolved? If yes specify the |

|  |  |
| --- | --- |
|  | allocation sequence, if no explain why?  b. Repeat part a. for resource r2.  c. Repeat part a. for resource r3. |
| 2 | What are the various methods for handling deadlocks? |
| 3 | What is a resource allocation graph? How do you obtain a wait-for graph from it? Explain their uses. |
| 4 | A system has four processes P0 through P3 and three resource types R1, R2 and R3. R1 has 2 units, R2 has 3 units and R3 has 2 units. Given that:  P0 holds 1 unit of R3 and R1 ; request 1 unit of R3 P1 holds 1 unit of R1 and request 1 unit of R1  P2 holds 1 unit of R2 and request 1 unit of R3  P3 holds 1 units of R2 ; request 2 unit of R3 and 1 unit of R1  Show the resource allocation graph for this state of the system. Is the system in deadlock, and if so, which processes are involved? |

|  |  |
| --- | --- |
| 1 | Consider the following two systems. Given that all resource instances are of the same type for each system:   1. Determine the values in the Need matrix for each process in the system. 2. Determine whether each of the systems is safe or unsafe. 3. If the system is in a safe state, list a sequence of requests and releases that will make it possible for   all processes to run to completion.   1. If the system is in an unsafe state, show how it is possible for deadlock to occur.   **System 1** has 12 instances of the resource. Only 1 is available.    **System 2** has 14 instances of the resource. 2 are available. |
| 2 | Describe the four conditions required for deadlock to occur. Describe a common method for deadlock prevention that prevents one of the conditions occurring. |

|  |  |
| --- | --- |
| 3 | Consider the following snapshot of a system:    Answer the following questions using the banker’s algorithm:   1. What is the content of the matrix Need? 2. Is the system in a safe state? 3. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately? |
| 4 | For which system Banker’s algorithm is applicable? Apply banker’s algorithm to the following snapshot taken at time t0 and find out process execution sequence that lands a system in safe state. |
| 5 | Explain Bankers’ algorithm to avoid deadlock. |
| 6 | Consider the following snapshot of a system:  Using Banker’s algorithm,   1. Is the system in a safe state? Justify, clearly showing the steps of the algorithm. 2. If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately? |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Allocation** | | | | **Max** | | | | **Total Instances** |
|  | **A** | **B** | **C** | **D** | **A** | **B** | **C** | **D** | **A B C D** |
| **P0** | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 2 | 1 5 2 0 |
| **P1** | 1 | 0 | 0 | 0 | 1 | 7 | 5 | 0 |  |
| **P2** | 1 | 3 | 5 | 4 | 2 | 3 | 5 | 6 |  |
| **P3** | 0 | 6 | 3 | 2 | 0 | 6 | 5 | 2 |  |
| **P4** | 0 | 0 | 1 | 4 | 0 | 6 | 5 | 6 |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | |  | | | |  |
|  |  | | | |  | | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |