

**INTERNAL ASSIGNMENT - JUL2023 **

INTERNAL ASSIGNMENT

**Course Code: OMC-109 Last Date of Submission: 15/01/24 Course Title: Operating Systems & Computer Networks lab Maximum Marks: 30**

**Session: July 2023**

Please submit answers to any four programs, accompanied by output snapshots. Ensure that at least two sets of outputs are provided for two different inputs. (6\*4=24)

|  |  |
| --- | --- |
| **Q.No.** | **Question** |
| 1 | Write a C program to Simulate the following Memory management algorithm-First fit  # include <stdio.h>  *void* first\_fit(*int* *m*, *int* *n*, *int* *Blocks*[], *int* *Process*[]){  *int* i,j;  *int* allocation[*n*];      for ( i = 0; i < *n*; i++)      {          /\* code \*/          allocation[i]= -1;      }      for ( i = 0; i < *n*; i++) // # processes      {          /\* code \*/          for (  j = 0; j<*m*; j++) // # blocks          {              /\* code \*/              if (*Blocks*[j] >=*Process*[i]){                  allocation[i]= j;  *Blocks*[j]= *Blocks*[j]-*Process*[i];                  break;                }          }        }      printf("\nP. No.\tP. Size\tBlock No.\n");      for (i = 0; i < *n*; i++)      {          /\* code \*/          printf("%d\t%d\t", i+1, *Process*[i]);          if (allocation[i]!=-1)          {              printf("%i\n", allocation[i]+1);          }else printf("Not Allocated\n");        }        }  *int* main(){  *int* m, n, Blocks[10], Process[10];      printf("Enter # processes: "); scanf("%d", &n);      printf("Enter # blocks: "); scanf("%d", &m);      printf("Enter the process sizes\n");      for (*int* i = 0; i < n; i++)      {          scanf("%d", &Process[i]);      }      printf("Enter the block sizes\n");      for (*int* i = 0; i < m; i++)      {          scanf("%d", &Blocks[i]);      }      first\_fit(m, n, Blocks, Process);      return 0;    } |
| 2 | Write a C program to Implement the optimal page replacement algorithm |
| 3 | Implement a program in C to extract process ID (PID) and parent process ID (PPID) |
| 4 | Simulate the following CPU scheduling algorithms-FCFS  #include<stdio.h>  #include<stdlib.h>  *struct* Process  {      /\* data \*/  *int* pid;  *int* bt;  *int* at;  };  *void* fcfs\_scheduling(*struct* Process\**proc*, *int* *n*){  *int* i, wt[*n*], tat[*n*], total\_wt=0, total\_tat=0;      // calculate waiting time for each process      wt[0]= 0;      for ( i = 1; i < *n*; i++)      {          /\* code \*/          wt[i]= wt[i-1]+ *proc*[i-1].bt;      }      // calculate turnaround time for each process      for ( i = 0; i < *n*; i++)      {          /\* code \*/          tat[i]= wt[i]+ *proc*[i].bt;      }      // calculate total waiting and turnaround time      for (i = 0; i < *n*; i++)      {          /\* code \*/          total\_wt+=wt[i];          total\_tat+=tat[i];      }      printf("\nPID\tBT\tAT\tWT\tTAT\n");      for ( i = 0; i < *n*; i++)      {          /\* code \*/          printf("%d\t%d\t%d\t%d\t%d\n", *proc*[i].pid, *proc*[i].bt, *proc*[i].at, wt[i], tat[i]);      }      printf("\nAverage waiting time: %.2f\n", (*float*)total\_wt/*n*);      printf("\nAverage turnaround time: %.2f\n", (*float*)total\_tat/*n*);      }  *int* main(){  *int* n, i;      printf("Enter the number of processes: ");      scanf("%d", &n);  *struct* Process proc[n];      for ( i = 0; i < n; i++)      {          /\* code \*/          printf("Enter the burst time and arrival time for process %d: ", i+1);          scanf("%d%d", &proc[i].bt, &proc[i].at);          proc[i].pid= i+1;      }      fcfs\_scheduling(proc, n);      return 0;    } |
| 5 | Write a C program to Implement the SSTF Disk scheduling |
| 6 | Implement the producer consumer problem with solution using semaphore |

Compulsory question: Explain the installation steps for Cisco Packet Tracer, and include snapshots for clarification. (6\*1=6)