Q.1 Write the simulation program to implement demand paging and show the page scheduling and total number of page faults according to the LFU page replacement algorithm. Assume the memory of n frames.

Reference String : 3,4,5,4,3,4,7,2,4,5,6,7,2,4,6 [15]

Q.2 Write a C program to implement the shell which displays the command prompt “myshell$”. It accepts the command, tokenize the command line and execute it by creating the child process. Also implement the additional command ‘typeline’ as

typeline +n filename :- To print first n lines in the file.

typeline -a filename :- To print all lines in the file. [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the FIFO page replacement algorithm. Assume the memory of n frames.

Reference String : 3, 4, 5, 6, 3, 4, 7, 3, 4, 5, 6, 7, 2, 4, 6 [15]

Q.2 Write a program to implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following ‘list’ commands as

myshell$ list f dirname :- To print names of all the files in current directory.

myshell$ list n dirname :- To print the number of all entries in the current directory [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program to implement demand paging and show the page scheduling and total number of page faults according to the LRU (using counter method) page replacement algorithm. Assume the memory of n frames.

Reference String : 3,5,7,2,5,1,2,3,1,3,5,3,1,6,2 [15]

Q.2 Write a programto implement the toy shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

count c filename :- To print number of characters in the file.
count w filename :- To print number of words in the file.
count l filename :- To print number of lines in the file. [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the MFU page replacement algorithm. Assume the memory of n frames.

Reference String : 8, 5, 7, 8, 5, 7, 2, 3, 7, 3, 5, 9, 4, 6, 2 [15]

Q.2 Write a program to implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search a filename pattern :- To search all the occurrence of pattern in the file.

myshell$ search c filename pattern :- To count the number of occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the optimal page replacement algorithm. Assume the memory of n frames.

Reference String : 8, 5, 7, 8, 5, 7, 2, 3, 7, 3, 5, 9, 4, 6, 2 [15]

Q.2 Write a program to implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search f filename pattern :- To display first occurrence of pattern in the file.

myshell$ search c filename pattern :- To count the number of occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

(

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the MRU page replacement algorithm. Assume the memory of n frames.

Reference String : 8, 5, 7, 8, 5, 7, 2, 3, 7, 3, 5, 9, 4, 6, 2 [15]

Q.2 Write a programto implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search f filename pattern :- To display first occurrence of pattern in the file.

myshell$ search a filename pattern :- To search all the occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

(

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the Optimal page replacement algorithm. Assume the memory of n frames.

Reference String : 7, 5, 4, 8, 5, 7, 2, 3, 1, 3, 5, 9, 4, 6, 2 [15]

Q.2 Write a program to implement shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search a filename pattern :- To search all the occurrence of pattern in the file.

myshell$ search c filename pattern :- To count the number of occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

Write the simulation program for demand paging and show the page scheduling and total number of page faults according the LRU page replacement algorithm. Assume the memory of n frames.

Reference String : 8, 5, 7, 8, 5, 7, 2, 3, 7, 3, 5, 9, 4, 6, 2 [15]

Q.1 Write a programto implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search f filename pattern :- To display first occurrence of pattern in the file.

myshell$ search c filename pattern :- To count the number of occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

(

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the FIFO page replacement algorithm. Assume the memory of n frames.

Reference String : 8, 5, 7, 8, 5, 7, 2, 3, 7, 3, 5, 9, 4, 6, 2 [15]

Q.2 Write a program to implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search f filename pattern :- To display first occurrence of pattern in the file.

myshell$ search a filename pattern :- To search all the occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]

(Write the simulation program for demand paging and show the page scheduling and total number of page faults according the FIFO page replacement algorithm. Assume the memory of n frames.

Reference String : 2, 4, 5, 6, 9, 4, 7, 3, 4, 5, 6, 7, 2, 4, 7, 1 [15]

Q.1 Write a program to implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following 'list' commands as

myshell$ list f dirname :- To print names of all the files in current directory.

myshell$ list i dirname :- To print names and inodes of the files in the current directory. [15]

Q.3. Oral/Viva [05]

.

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the LFU page replacement algorithm. Assume the memory of n frames.

Reference String : 3, 4, 5, 6, 3, 4, 7, 3, 4, 5, 6, 7, 2, 4, 6 [15]

Q.2 Write a C program to implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following ‘list’ commands as

myshell$ list f dirname :- To print names of all the files in current directory.

myshell$ list n dirname :- To print the number of all entries in the current directory [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the LRU page replacement algorithm. Assume the memory of n frames.
Reference String : 3, 4, 5, 6, 3, 4, 7, 3, 4, 5, 6, 7, 2, 4, 6 [15]

Q.2 Write a program to implement the shell. It should display the command prompt "myshell$". Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following 'list' commands as

myshell$ list f dirname :- To print names of all the files in current directory.

myshell$ list n dirname :- To print the number of all entries in the current directory [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to implement the shell which displays the command prompt “myshell$”. It accepts the command, tokenize the command line and execute it by creating the child process. Also implement the additional command ‘typeline’ as

typeline -a filename :- To print all lines in the file. [15]

Q.2 Write the simulation program for Round Robin scheduling for given time quantum. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give the Gantt chart, turnaround time and waiting time for each process. Also display the average turnaround time and average waiting time. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to implement the shell which displays the command prompt “myshell$”. It accepts the command, tokenize the command line and execute it by creating the child process. Also implement the additional command ‘typeline’ as

typeline +n filename :- To print first n lines in the file. [15]

Q.2 Write a C program to simulate Non-preemptive Shortest Job First (SJF) – scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time[15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following ‘list’ commands as

myshell$ list f dirname :- To print names of all the files in current directory.

[15]

Q.2 Write the program to simulate preemptive Shortest Job First (SJF) – scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time[15]

Q.3. Oral/Viva [05]

Q.1 Write a programto implement the toy shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

count c filename :- To print number of characters in the file.

count w filename :- To print number of words in the file. [15]

Q.2 Write the program to simulate Non preemptive priority scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output

should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time.[15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the Optimal page replacement algorithm. Assume the memory of n frames.

Reference String : 7, 5, 4, 8, 5, 7, 2, 3, 1, 3, 5, 9, 4, 6, [15]

Q.2 Write the program to simulate FCFS CPU-scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time. [15]

Q.3. Oral/Viva [05]

Q.1 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the LRU page replacement algorithm. Assume the memory of n frames.

Reference String : 3, 4, 5, 6, 3, 4, 7, 3, 4, 5, 6, 7, 2, 4, 6 [15]

Q.2 Write a C program to simulate FCFS CPU-scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following ‘list’ commands as

myshell$ list f dirname :- To print names of all the files in current directory. [15]

Q.2 Write the simulation program for Round Robin scheduling for given time quantum. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give the Gantt chart, turnaround time and waiting time for each process. Also display the average turnaround time and average waiting time. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to implement the shell which displays the command prompt “myshell$”. It accepts the command, tokenize the command line and execute it by creating the child process. Also implement the additional command ‘typeline’ as

typeline -a filename :- To print all lines in the file. [15]

Q.2 Write the program to simulate Non-preemptive Shortest Job First (SJF) – scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time[15]

Q.3. Oral/Viva [05]

Q.1 Write a C Program to create a child process using fork (), display parent and child process id. Child process will display the message “I am Child Process” and the parent process should display “I am Parent Process”. [15]

Q.2 Write a C program to simulate Preemptive Priority scheduling. The arrival time and first CPU-burst and priority for different n number of processes should be input to the algorithm. Assume the fixed IO waiting time (2 units). The next CPU-burst should be generated randomly. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program that demonstrates the use of nice() system call. After a child Process is started using fork (), assign higher priority to the child using nice () system call. [15]

Q.2 Write a C program to simulate Non preemptive priority scheduling. The arrival time and first CPU-burst of different jobs should be input to the system. Accept no. of Processes, arrival time and burst time. The output should give Gantt chart, turnaround time and waiting time for each process. Also find the average waiting time and turnaround time. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to illustrate the concept of orphan process. Parent process creates a child and terminates before child has finished its task. So child process becomes orphan process. (Use fork(), sleep(), getpid(), getppid()).

[15]

Q.2 Write the simulation program for demand paging and show the page scheduling and total number of page faults according the Optimal page replacement algorithm. Assume the memory of n frames.

Reference String : 7, 5, 4, 8, 5, 7, 2, 3, 1, 3, 5, 9, 4, 6, [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program to accept n integers to be sorted. Main function creates child process using fork system call. Parent process sorts the integers using bubble sort and waits for child process using wait system call. Child process sorts the integers using insertion sort. [15]

Q.2 Write a C program to implement the toy shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

count c filename :- To print number of characters in the file.
count w filename :- To print number of words in the file.
count l filename :- To print number of lines in the file. [15]

Q.3. Oral/Viva [05]

Q.1 Write a C program that accepts an integer array. Main function forks child process. Parent process sorts an integer array and passes the sorted array to child process through the command line arguments of execve() system call. The child process uses execve() system call to load new program that uses this sorted array for performing the binary search to search the particular item in the array. [15]

Q.2 Write a programto implement the shell. It should display the command prompt “myshell$”. Tokenize the command line and execute the given command by creating the child process. Additionally it should interpret the following commands.

myshell$ search f filename pattern :- To display first occurrence of pattern in the file. [15]

Q.3. Oral/Viva [05]