1.Write a program to implement an address book with options given below:

a) Create address book. b) View address book. c) Insert a record. d) Delete a record. e) Modify a record. f) Exit

2.A. Process control system calls: The demonstration of FORK, EXECVE and WAIT system calls along with zombie and orphan states.

Implement the C program in which main program accepts the integers to be sorted. Main program uses the FORK system call to create a new process called a child process. Parent process sorts the integers using sorting algorithm and waits for child process using WAIT system call to sort the integers using any sorting algorithm. Also demonstrate zombie and orphan states.

2.B. Implement the C program in which main program accepts an array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an 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 which display array in reverse order.

3. Implement the C program for CPU Scheduling Algorithms: Shortest Job First

(Preemptive) and Round Robin with different arrival time.

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| A | 0 | 5 |
| B | 2 | 4 |
| C | 2 | 2 |
| D | 4 | 4 |

4.A. Thread synchronization using counting semaphores. Application to demonstrate: producer consumer problem with counting semaphores and mutex.

4.B. Thread synchronization and mutual exclusion using mutex. Application to demonstrate: Reader Writer problem with reader priority.

5. Implement the C program for Deadlock Avoidance Algorithm: Bankers Algorithm.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Process no | Allocation | | | | Max | | | | Available | | | | | Need | | | | |
|  | A | B | C | D | 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 | |  |  |  |  | |  |  |  |  |

6. Implement the C program for Page Replacement Algorithms: FIFO, LRU, and Optimal for frame size as minimum three.

Consider the following page reference string:

1. 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

7.A. FIFOS: Full duplex communication between two independent processes. First process accepts sentences and writes on one pipe to be read by second process and second process counts number of characters, number of words and number of lines in accepted sentences, writes this output in a text file and writes the contents of the file on second pipe to be read by first process and displays on standard output.

7.B. Inter-process Communication using Shared Memory using System V. Application to demonstrate:

Client and Server Programs in which server process creates a shared memory segment and writes the message to the shared memory segment. Client process reads the message from the shared memory segment and displays it to the screen.

8. Implement the C program for Disk Scheduling Algorithms: SSTF, SCAN, C-Look

considering the initial head position moving away from the spindle.