2018

CBCS

3rd Semester

COMPUTER SCIENCE

PAPER—C6P

(Set-1)

(Honours)

(Practical)

Full Marks : 20

Time : 2 Hours

The figures in the right-hand margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Illustrate the answers wherever necessary.

Operating Systems Lab.

Answer any *one* question using C / C++ Language *(Lottery basis)* : 1×15

1. Write a program to fork a new child process and parent process should wait for the completion of child.

(Turn Over)

2. Write a program to fork a new child process where parent and child execute same program and same code.

3. Write a program to print details (owner access permission, file access time) of an input file.

4. Write a program to copy files using system calls.

5. Write a program to create a new process using system ( ) that displays the processes running on your system.

6. Write a program to implement worst-fit allocation strategy.

7. Write a program to calculate sum of *n* numbers using thread library.

8. Write a C/C++ Program to copy a file using read ( ) and write ( ) system calls.

9. Write a C/C++ Program to display file type as 'Regular' or 'Directory' or 'other' of the file whose name is obtained from command line argument.

10. Write a C/C++ program to demonstrate interprocess communication between two processes.

11. Write a C/C++ program to create two processes and kill them afterwards using signals.

12. Write a C/C++ program to implement Round Robin CPU scheduling algorithm.

[PNB – 02; Viva Voce – 03]

Total Pages—3 **C/18/BSc/3rd Sem/COSH/C6P(S2)**

2018

CBCS

3rd Semester

COMPUTER SCIENCE

PAPER—C6P

(Set–2)

(Honours)

(Practical)

Full Marks : 20

Time : 2 Hours

The figures in the right-hand margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Illustrate the answers wherever necessary.

Operating Systems Lab.

Answer any *one* question using C / C++ Language *(Lottery basis)* : 1×15

1. Write a program to the post behaviour of Linux Kernel including information on configured memory, amount of free and used memory (memory information).

(Turn Over)

2. Write a program to fork a new child and parent should wait for the completion of child.

3. Write a program to implement best-fit allocation strategy.

4. Write a program to implement FCFS scheduling algorithm.

5. Write a program to create a new process using system ( ) that display list of files.

6. Write a program to implement worst-file allocation strategy.

7. Write a program to copy files using system calls.

8. Write a C/C++ Program to read content of a file using read ( ) system call and display the content in monitor.

9. Write a C/C++ Program to read a file name as command line argument and enable it's 'execution' Permission for 'owner' and 'group' user.

10. Write a C/C++ program to display all the environment variables in your system.

11. Write a C/C++ program to implement SJF CPU scheduling algorithm.

12. Write a C/C++ program to create a child process and then implement interprocess communication between parent and child process using semaphore.

13. Write a C/C++ program to implement interprocess communication between parent and child process using pipe.

14. Write a C/C++ program to create a child process such that both the parent and child process execute same program, same code and parent must wait for the child to terminate.

[PNB – 02; Viva Voce – 03]