9/12/19 (M)

(74)

[This question paper contains 12 printed pages]

Your Roll No. : .....

Sl. No. of Q. Paper : **7404** **J**

Unique Paper Code : 32341302

Name of the Course : **B.Sc.(Hons.) Computer Science**

Name of the Paper : Operating Systems

Semester : III

Time : 3 Hours ***Maximum Marks : 75***

Instructions for Candidates :

(a) Write your Roll No. on the top immediately on receipt of this question paper.

(b) Question No. **1** of **35** marks is compulsory.

(c) Attempt any **four** questions from Question No. **2** to Question No. **7**.

1. (a) Fill in the blanks : 6

(i) ..... is a necessary condition for a deadlock according to which at least one resource is held in a non-sharable mode.

P.T.O.

(ii) Loading the pages of a process into memory only when they are needed is termed as ..... .

(iii) Updating the caches of all processors to reflect any modification of data in one cache is termed as ..... .

(iv) The time needed for the required sector to rotate to the disk head during a disk access is termed as ..... .

(v) ..... is the location within the directory structure where a file system is to be attached (in Unix system).

(vi) ..... provide an interface to the services made available by an operating system.

(b) Differentiate between : 2×4=8

(i) zombie and orphan process

(ii) mutex and binary semaphore

(iii) system and application program

(iv) symmetric and asymmetric multiprocessing

(c) What will be the output of the following code ? Explain your answer. 3

```
int main()
{
  int x = 1, p;
  p = fork();
  if(p == 0)
    x = 10;
  else
  {
    wait(NULL);
    printf("%d\n",x);
  }
}
```

(d) Given the logical address 0xAEF9 (in hexadecimal) with a page size of 256 bytes, determine (i) the page number (ii) page offset. 3

(e) What is file-open count ? Where is it stored ? When does its value become zero ? 3

(f) What will be the output of the following code ? 3

```
int main()
{
  pid_t pid;
  execlp("/bin/ls","ls", NULL);
  pid = fork();
  if(pid<0)
  {
    printf("fork failed");
    return 1;
  }
  else if(pid==0)
    execlp("/bin/ls","ls",NULL);
  else
  {
    wait(NULL);
    printf("child finished");
  }
  return 0;
}
```