BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI (RAJASTHAN)
First Semester, 2019-2020
Quiz

Course Number: CS F372 **Course Title:** Operating Systems **Date and Time:** Nov 08, 2019 (08.00 AM To 08.50 AM)
Weightage & Nature : 20 % [Open Book] **Maximum marks:** 40

ID.No. 2017A7PS0004P **Name:** Subham Kumar Dash.

Instructions: Please write answer for each question in the space provided in the answer-sheet. Please do not write anything else in question paper other than the required entries. There is no negative marking. While answering the questions, use the constructs and syntax applicable to UNIX System V only.

Q1. For each subpart, two statements, namely S1 and S2, are given. Answer (in BLOCK CAPITAL letters) as A if S1 alone is true, as B if S2 alone is true, C if both S1 and S2 are true, and as D otherwise.

1.1 S1: Message queue is useful for exchanging bigger amount of data, because no conflicts need to be avoided. S2: In fork() system call, the parent and child processes share only the shared memory segments while stack and heap are duplicated. A

1.2 S1: An allocated data block can belong to one and only one file in the file system. S2: Message queue scheme provides a mechanism to allow processes to communicate and to synchronize their actions without sharing the same address space. B

1.3 S1: It is the inode by which the directory structure of the file system is accessible even before the execution of mount system call. S2: The directory structure used in Unix file system is called Directed acyclic graph.

1.4 S1: The default permission bits of a file when it is created for the first time, is controlled by chmod value. S2: In dup2(), if oldfd has the same value as newfd, oldfd is silently closed before being duplicated.

1.5 S1: In the message queue scheme, communication takes place by means of messages exchanged between the cooperating processes. S2: Special file used to represent a real physical device such as a printer, tape drive or terminal, used for I/O operations.

Marks Q1 [1 x 5 = 5]

Q2. Fill-in the blanks.

2.1 A process loaded by kernel consists of three parts text, data and _____.

2.2 After a fork() system call, one of the process typically uses the _____ system call to replace the process's memory space with a new program.

2.3 To increase the response time and throughput, the kernel minimizes the frequency of disk access by keeping a pool of internal data buffer called _____.

2.4 To run the script, we should make it executable first by using chmod _____.

2.5 _____ IPC scheme allows for communication between two unrelated process in full duplex mode.

Marks Q2 [1 x 5 = 5]

Q3. Write the output of the following program segments. Assume that appropriate header files have been included.

3.1

```
#include <stdio.h>
#include <unistd.h>
int main()
{
  int i = 0;
  if(fork() == 0) ++i;
  else ++i;
  printf("%d ", ++i);
  return 0;
}
```

3.2

```
int main()
{
    if (fork() || fork())
        if (fork() && fork())
            fork();
    printf("F ");
    return 0;
}
```

3.3

```
int main()
{
int i, n=3;
for (i = 0; i < n; i++)
        { fork(); }
printf("F ");
return 0;
}
```

3.4

```
int main() {
  int pid, pid1;
  pid = fork();
  if (pid == 0) {
    sleep(1);
    printf("I am child [%d] \n", 1);
  }
  else {
    pid1 = fork();
```

```
    if (pid1 == 0) {
      printf("I am child [%d] \n", 2);
    }
    else {
      sleep(2);
      printf("I am parent \n");
    }
  }
  return 0;
}
```

3.5

```
int main()
{
    int fd1,fd2,fd3,fd4;
    fd1 = open("t1.txt",O_RDONLY|O_CREAT,0777);
    fd2 =  open("t2.txt",O_RDONLY|O_CREAT,0777);
    close(fd1);
    fd3 = open("t3.txt",O_RDONLY|O_CREAT,0777);
    fd4 = open("t4.txt",O_RDONLY|O_CREAT,0777);
    printf("opened fd_2  = % d\n",fd2);
    printf("opened fd_3 = % d\n",fd3);
    printf("opened fd_4 = % d\n",fd4);
}
```

Marks Q3 [2 x 5 = 10]

Q4. Solve the following:

4.1 Given the file size field in the inode = 32 bit, Logical block size = 8K byte, if the sizes of the files are such that the number of bytes in the last block of a file is uniformly distributed in [2, 4096] bytes, calculate the average wasted space per file.

4.2 Assume that in a directory called parentDir (owned by users group) there are 10 regular files (with each file size not exceeding 8KB and each having at most 99 links) and two empty subdirectories named dir1 and dir2. Assume that block size is 4KB. These files and subdirectories are owned by user user1. How many maximum number of characters will be displayed (between the two command prompts) by the execution of the command 'ls –l' issued after changing the working directory to the parentDir?

Marks Q4 [2 + 2 = 4]

Q5. The following program is expected to behave as if the Parent and Child do not share file access. The program also copies (byte by byte) the contents of one file into the other file. Complete the program by filling up text appearing a "**Q5.1**" to "**Q5.6**".

```
#include <  Q5.1  >
int fdR; /* File descriptor of the file opened for reading,
filename given as second argument  */
 int fdW; /* File descriptor of the file opened for writing,
filename given as third argument */
char ch;
main(int argc, char *argv[])
 {
   if (argc != 3) exit(1);
   Q5.2 ;
   if (fdR = open(  Q5.3 , O_RDONLY) == -1)
      exit(1);
   if (fdW = create(argv[2],  Q5.4 ) && ((fdW =
Q5.5(argv[2], O_WRONLY)) == -1)
      exit(1);
   ReadWrite();
}
ReadWrite()
{
 for(;;)
  {
    if (read(fdR, &ch, 1)   Q5.6   1)
      return;
   write(fdW, &ch, 1);
  }
}
```

Marks Q5 [1 x 6 = 6]

Q6. Complete (by filling missing statements i.e. Q6.1 to Q6.10) the following code snippet for copying contents of file text_1.txt to the file text_2.txt in reverse order using lseek() system call. **NOTE:** single character can be used for read/write operation.

```
/* Required header files */
_____                                    //(Q6.1)
int main() {
    int fd1, fd2; char c;  int offset;
    fd1 = open("text_1.txt", _______);       //(Q6.2)
    fd2 = open("text_2.txt", _______);       //(Q6.3)
    //write lseek command to go the end of text_1
    pos = _____;                             //(Q6.4)
    //while it is not the beginning of the file
    while (pos > 0) {
        _____;                               //(Q6.5)
        _____;                               //(Q6.6)
        _____;                               //(Q6.7)
        _____;                               //(Q6.8)
    }
    //close opened files
    _____;                                   //(Q6.9)
    _____;                                   //(Q6.10)
    return 0;
}
```

Marks Q6 [10]