Dhirubhai Ambani Institute of Information & Communication Technology
Second In-Semester Examination, Winter Semester 2022-2023

Course Title	IT628 Systems Programming	Max Marks	50
Date	28th Mar 2023	Time	1.5 Hours

Q1a(2)	Q1b(2)	Q1c(2)	Q1d(2)	Q1e(2)	Q1f(2)	Q1g(2)	Q1h(2)	Q1i(2)	Q1j(2)	Q1k(2)	Q2a(8)	Q2b(6)	Q3a(7)	Q3b(7)
2	0	2	1	0	1	1	0	0	0.5	0	2	0	2	0

Instructions:

1. All questions are compulsory.
2. Write the answers for all questions in the space provided in question paper only.
3. Write your answers in brief, writing long answers does not fetch higher marks.

Q1. Short Answer 1-3 lines [2 Marks each]

a. Upon executing fork(), code and data area of parent and child processes look the same however context look different. Explain briefly why?

Answer: Context is different because :- Different process ids of parent and child. Parent's PPID is different and child's PPID is as same as parent's process ID.

2

b. Once the following code is executed, specify how many entries will be created in system-wide file open table and what will be the values of offset and reference count for each entry. Assume that file1.txt exist with the file size of 100 bytes.

```
char buf[100];
int fd1=open("file1.txt", "O_RDWR ");
int fd2=open("file1.txt", "O_RDWR ");
write(fd1 , buf, 10);
read(fd2, buf, 20);
```

Answer:

file descriptor	offset	reference count
01	1	2
01	0	2

c. What will be the output from the code below? Explain briefly. (Extra space on the next page available)

```
int x=10;
printf("%d\n",x);
fork();
int y=x + 5;
printf("%d %d\n",x,y);
```

Answer:

```
Output ⇒
10
10 15
10 15
```

2

Reason ⇒ because fork has created two process. One is the parent process itself and it's child process. So the printf statement will execute (one from two times the printf statement will execute (one from parent and another one from child process)

d. **When performing "ls -i" command in current directory following content is displayed.**

```
1000 file1.txt
1000 file2.txt
1001 file3.txt -> file1.txt
```

We execute the three commands "rm file1.txt", "cat file2.txt", "cat file3.txt" in the same order. What will be the output from these commands? Explain in brief.

Answer: 1st command will remove the file1.txt 1

2nd command and 3rd command will display the content of the files.

e. **What will be the output from the code below? Explain briefly.**

```
int x=10;
printf("%d\n",x);
if (fork() == 0)
   int y = 5;
int z=x+y;
printf("%d %d %d\n",x,y,z);
```

Answer:

output :

10

5 [if process is a child process otherwise 5 would not be displayed if the process is a parent process]

Compile time error!

10 5 15 [this result will be displayed if the process is a child process]

f. **What is the difference between /bin and /usr/bin directories? Briefly explain with an example why both are required.** 1

Answer: Difference is of the storage path and user related files will be stored in /usr/bin and other user related permissions aswell. Both are needed be if the files or any searched thing is not in the /usr/bin directory then it will look for the required files in /bin directory

g. Considering the code below what are sequence of states a child process goes through while reading a complete file.txt? Assume that file.txt has 15 bytes of data and possible process lifecycle states are idle, runnable, running, sleeping, suspended and zombified.

1

```
int fd = open("file.txt", O_RDONLY);
if (fork() == 0) {
    int bytes_read=100;
    while (bytes_read>=10)
        bytes_read = read(fd, buf, 10);
}
```

Answer: States will be ⇒ idle, runnable, running, ~~suspended~~.

→ Running (or) idle.

Explain briefly the output from the code assuming that file.txt has "-r--r--r--" permission.

0

```
struct stat st;
stat("file.txt", &st);
printf("%d", st.st_mode & S_IRUSR);
```

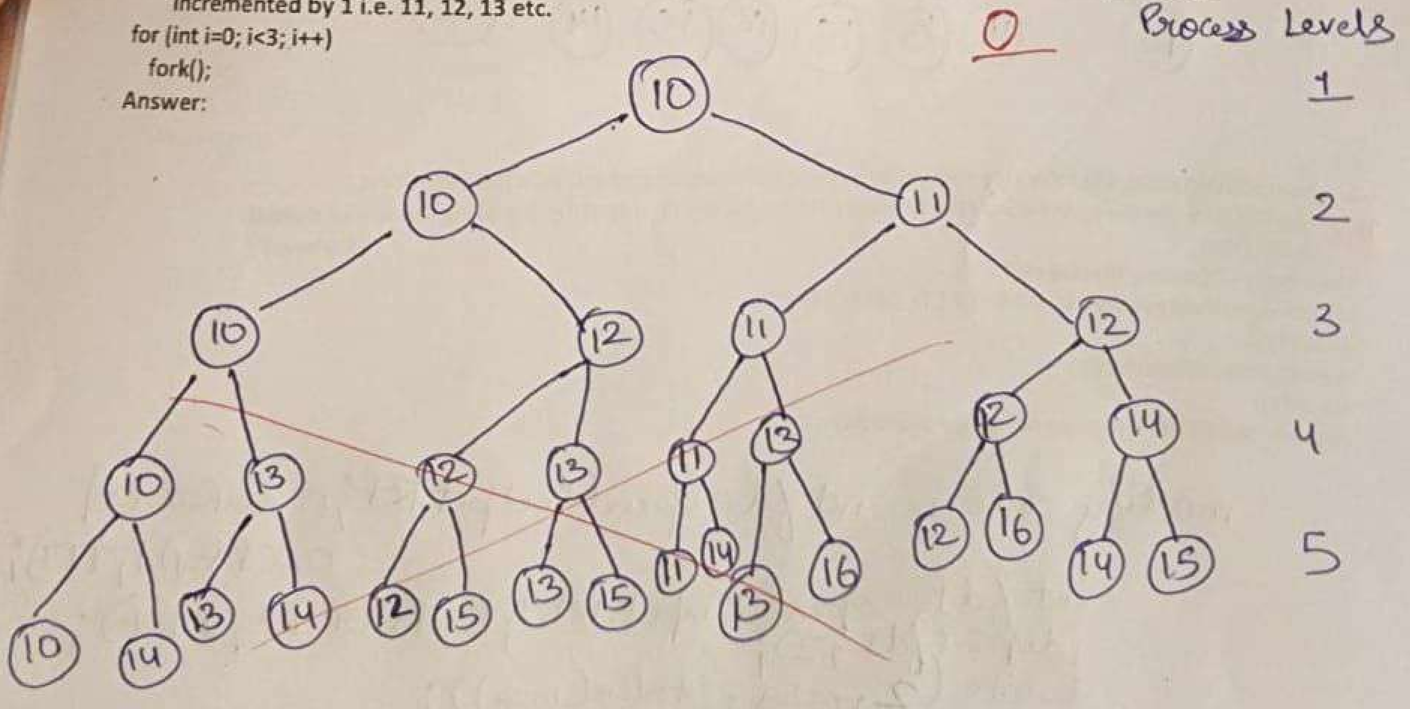
Answer: The output will be the contents of the file stored ~~which~~ which will can only be read and displayed to the user.

i. Considering a code below, draw a binary tree to show how many processes will be created? Assume that process id of main process starts with 10 and each subsequently created processes have process ids incremented by 1 i.e. 11, 12, 13 etc.

0

```
for (int i=0; i<3; i++)
    fork();
```

Answer:



j. Once the following code is executed, specify how many entries will be created in system-wide file open table and what will be the values of offset and reference count for each entry. Assume that file1.txt exist with the file size of 100 bytes.

0.5

```
char buf[100];
int fd1=open("file1.txt", "O_RDWR ");
int fd2=dup(fd1);
read(fd1 , buf, 20);
write(fd2, buf, 10);
```

Answer:

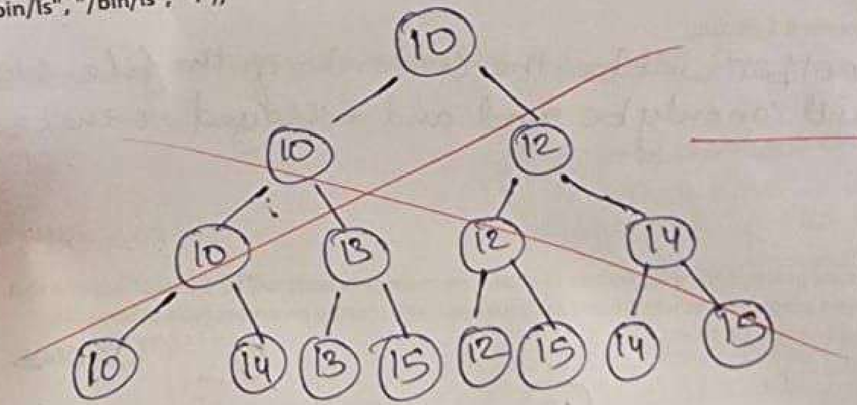
file descriptor	offset	reference count
01	0 1 2	2

k. Considering a code below, draw a binary tree to show how many processes will be created? Assume that process id of main process starts with 10 and each subsequently created processes have process ids incremented by 1 i.e. 11, 12, 13 etc.

0

```
for (int i=0; i<2; i++) {
  fork();
  execl("/bin/ls", "/bin/ls", "-l");
}
```

Answer:



Q2

a. As studied in class, the following code redirects standard output (stdout) to a file output.txt. Modify/Rewrite the code to copy the content of input.txt file to output.txt file using input and output redirection. [8 Marks]

2

```
char msg[] = "Dummy Message\n";
int fd = open("output.txt", O_WRONLY | O_CREAT, 777);
dup2( fd, 1);
write(1, msg, strlen(msg));
close(fd);
```

Answer: (Additional Space on next page available)

modify code → int fd = open("output.txt", O-WRONLY|
O-CREAT, 777);

```
int fd1 = open("input.txt", O-RDONLY, 777);
dup2(fd1, 2);
write(2, msg, strlen(msg));
close(fd);
```

use loop for R/W

dup2(fd, 1);

b. Considering File Control Block (FCB) or inode supports 10 direct pointers, 10 single indirect pointers, 5 double indirect pointers and 2 triple indirect pointers. Calculate the maximum size of the file supported by this FCB when each of the data block address is 32 bits. Assume that each data block is of size 4KB (4096 bytes). [6 Marks]

0

Answer: Maximum size of the file would be ⇒ 536976 bytes

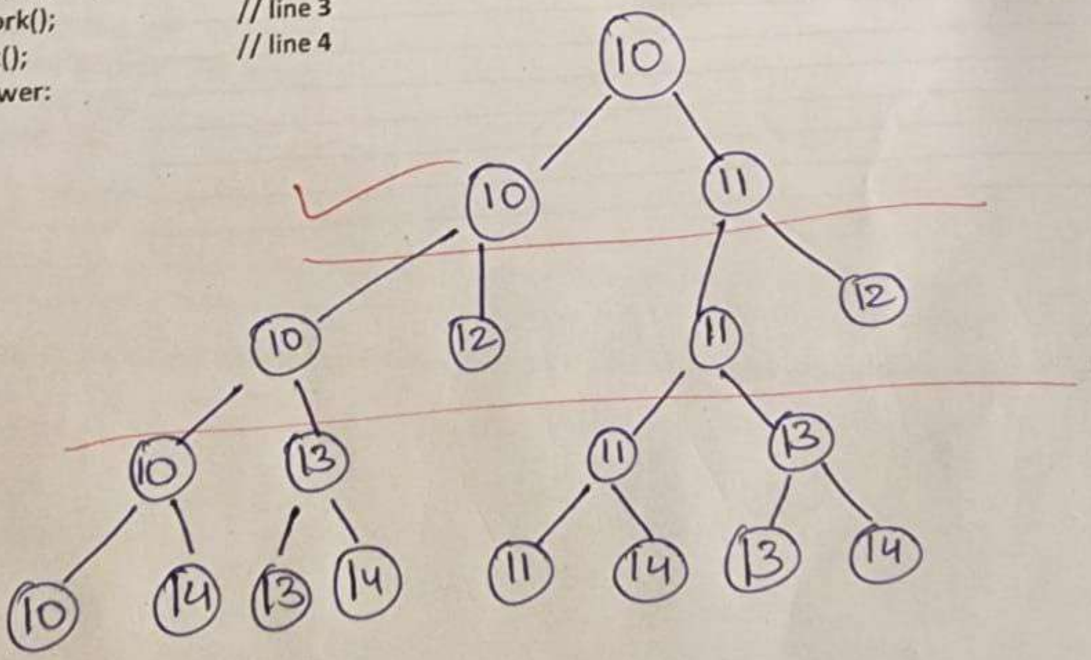
Q3

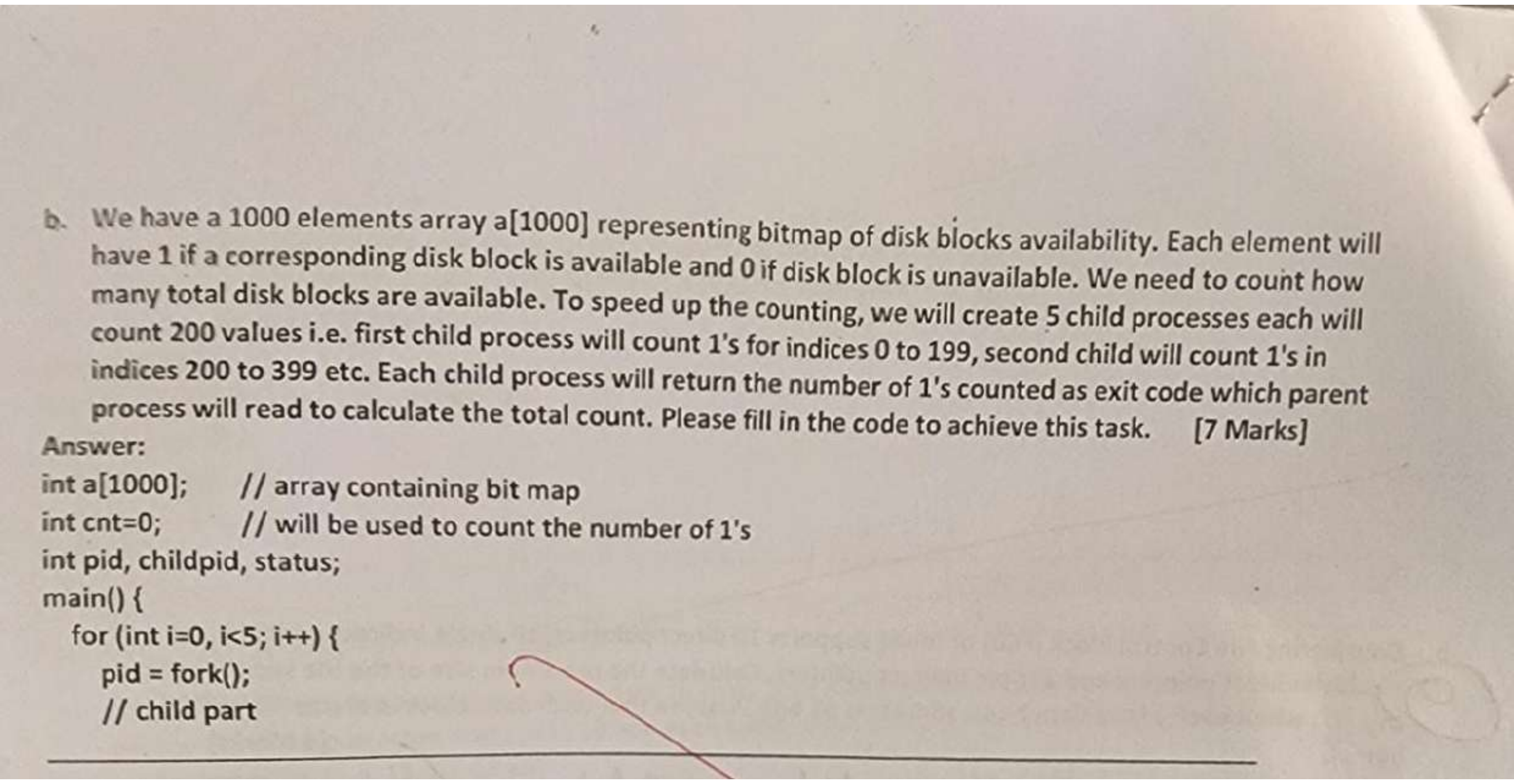
a. Considering a code below, draw a binary tree to show how many processes will be created? Assume that process id of main process starts with 10 and each subsequently created processes have process ids incremented by 1 i.e. 11, 12, 13 etc. [7 Marks]

```
fork();              // line1
if (fork() > 0)      // line2
  fork();            // line 3
fork();              // line 4
```

2

Answer:





b. We have a 1000 elements array a[1000] representing bitmap of disk blocks availability. Each element will have 1 if a corresponding disk block is available and 0 if disk block is unavailable. We need to count how many total disk blocks are available. To speed up the counting, we will create 5 child processes each will count 200 values i.e. first child process will count 1's for indices 0 to 199, second child will count 1's in indices 200 to 399 etc. Each child process will return the number of 1's counted as exit code which parent process will read to calculate the total count. Please fill in the code to achieve this task. [7 Marks]

Answer:

```
int a[1000];      // array containing bit map
int cnt=0;        // will be used to count the number of 1's
int pid, childpid, status;
main() {
  for (int i=0, i<5; i++) {
    pid = fork();
    // child part
```