Lebanese University
Faculty of Sciences I

6 April 2016
Duration : 1 hour

INFO 324
Operating System II

Dr. Ahmad Faour

Problem I

6 points

Draw the graph generated by the following code:

```
void main(){
        if((fork() || fork()) && !(fork()))
                fork();
}
```

Problem II

12 points

Giving the following program:

```
1.    void main(){
2.            int p[2], x = 1;
3.            pipe(p);
4.            if(fork()){
5.                    printf("A\n");
6.                    if(fork(){
7.                            // Ligne 7 wait(0) write(P[1]
8.                            // Ligne 8 wait(0) ~~[illegible]~~
9.                            printf("B\n");
10.                   }
11.                   else{
12.                           // Ligne 12 read(P[0],--- )
13.                           printf("C\n");
14.                   }
15.           }
16.           else{
17.                   // Ligne 17   read(P[0],---
18.                   printf("D\n");
19.                   // Ligne 19   write(P[1],---
20.           }
      }
```

The following questions are independent

a) What are the possible outputs?

b) We add the following statements :
```
Line 7 : wait(0) ;
Line 8 : wait(0) ;
```
What are the possible outputs?
Justify your answer.

c) Now, we add the following statements :
```
Line 7 : write(p[1],&x,sizeof(int));
Line 8 : write(p[1],&x,sizeof(int));
Line 12 : read(p[0],&x,sizeof(int));
Line 17 : read(p[0],&x,sizeof(int));
```
What are the possible outputs?
Justify your answer.

d) We add also the following :
```
Line 7 : write(p[1],&x,sizeof(int));
Line 12 : read(p[0],&x,sizeof(int));
Line 17 : read(p[0],&x,sizeof(int));
Line 19 : write(p[1],&x,sizeof(int));
```
What are the possible outputs? Justify your answer.

Problem III

12 points

1. Write a program that creates N child processes. The child processes shared a pipe of communication, and behave according to the following :

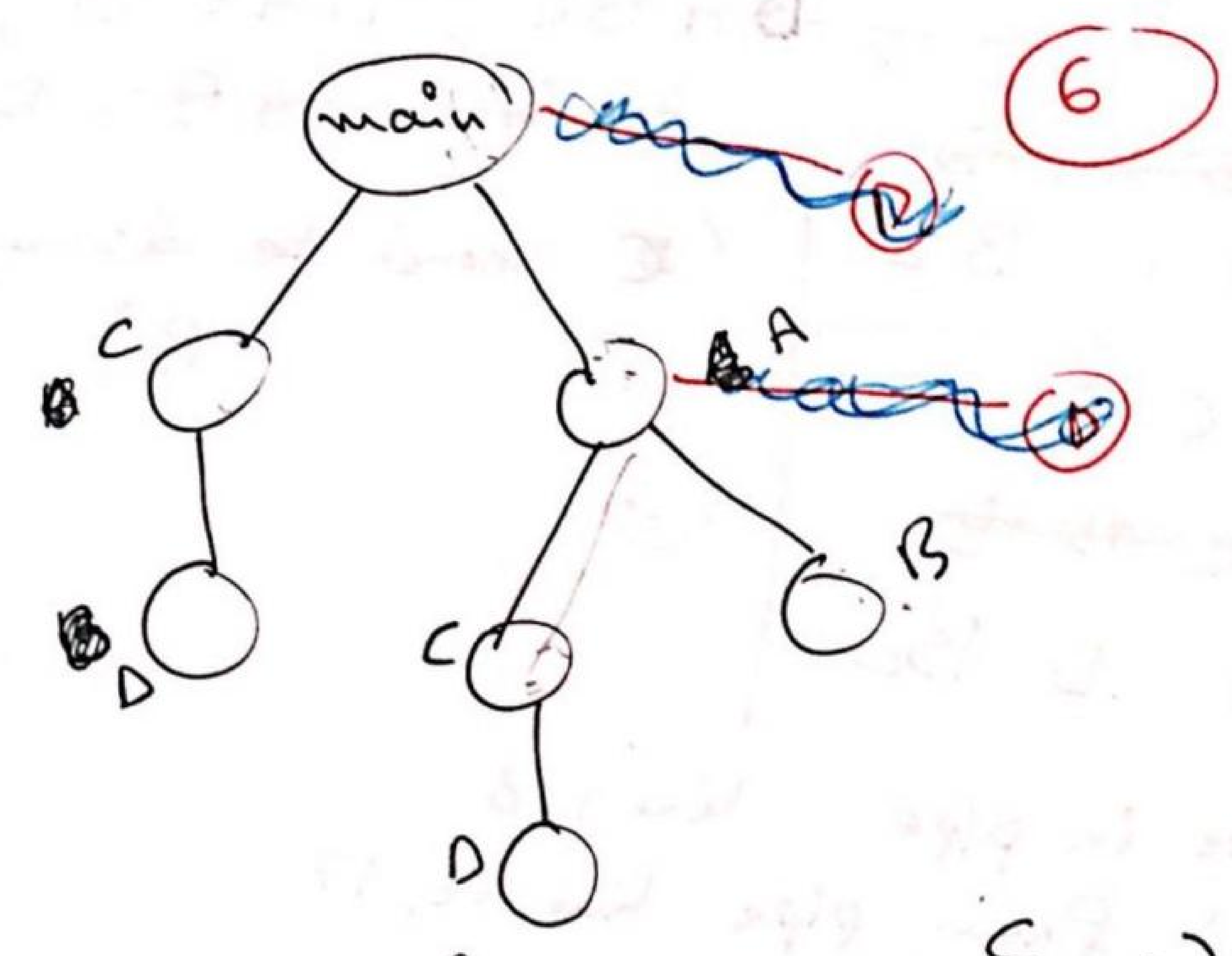
   - The child processes with even id (i.e., 0th process, 2th process...) are producers of messages: each one wait a time t before writing in the pipe the character 'a' and do the same indefinitely.
   - The child processes with odd order (1th process, 3th process...) are consumers of messages: each one wait a time t before reading from the pipe character by character and in indefinitely manner.

2. Now, suppose that the writers are faster than readers. Rewrite the program such that the pipe does not contain more than M characters (i.e., if the pipe contains M characters, no any process can write in it). The readers can read from pipe as usual.

<u>**P.S:**</u> think of the way to communicate the variant number of characters inside the pipe between processes (i.e., the reader and writer processes must be up to date regarding the contents of the pipe).

Good Luck

Ex1

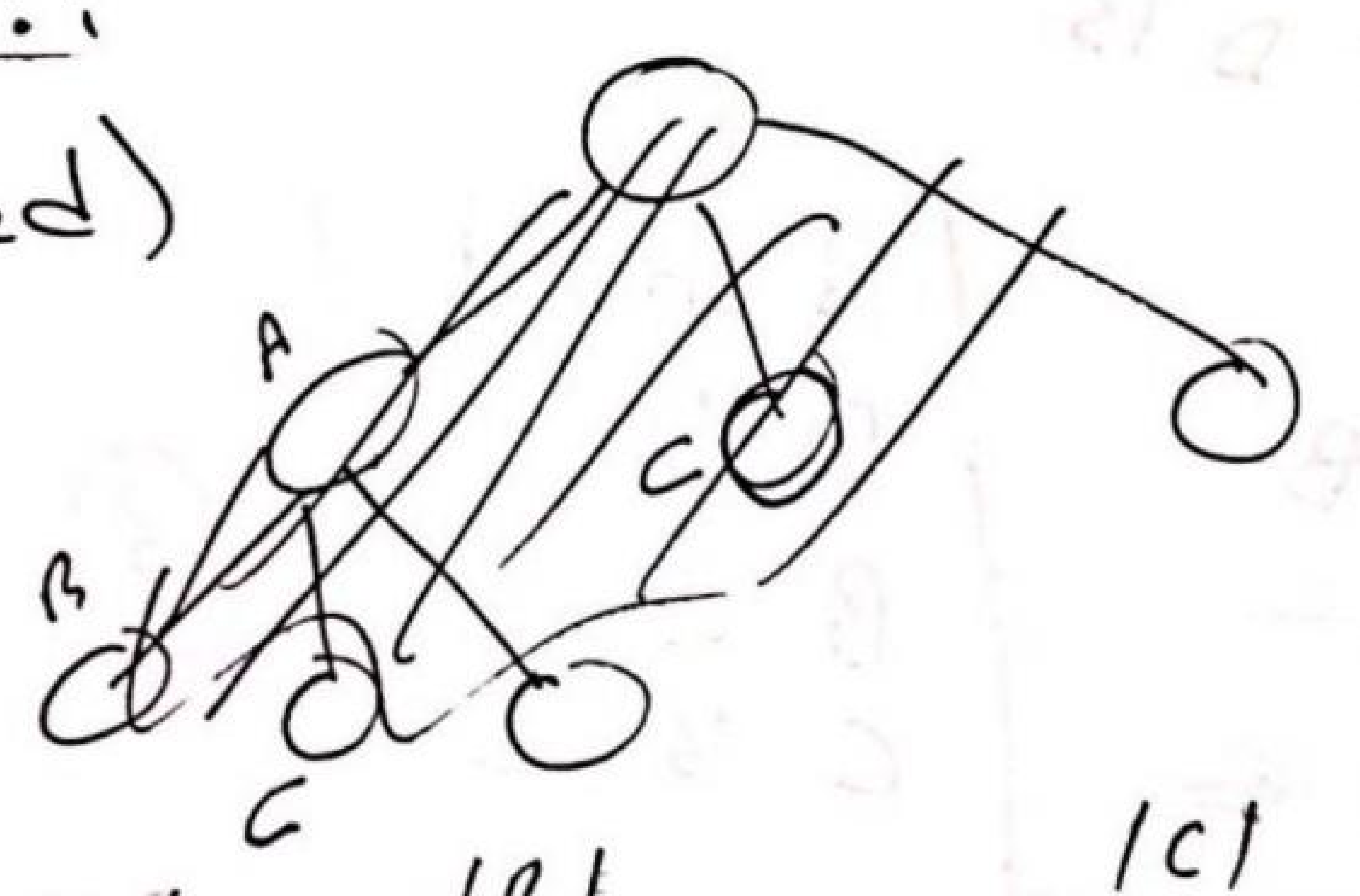


$\text{if } ((\overset{A}{\text{fork()}} \ || \ \overset{B}{\text{fork()}}) \ \&\& \ !(\overset{C}{\text{fork()}}))$

$\overset{D}{\text{fork()}};$

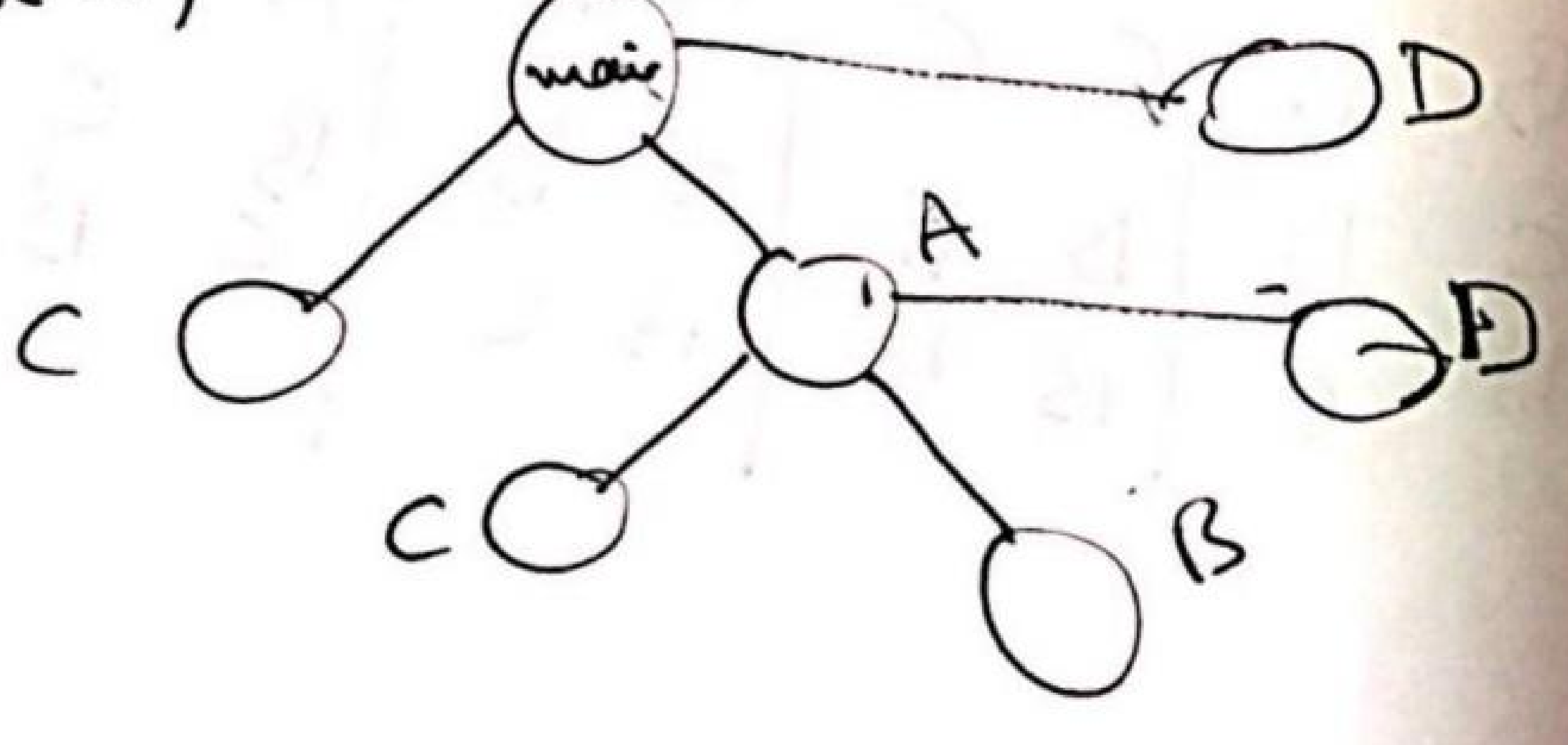
without !:

(Not requested)



$\text{if } ((\overset{/A/}{\text{fork()}} \ || \ \overset{/B/}{\text{fork()}}) \ \&\& \ \overset{/C/}{\text{fork()}})$

$\overset{/D/}{\text{fork()}};$



EX2 :

a) ~~A B D C , B A B C , D A C B , A D C B~~
~~A D B C~~ ~~A B C D~~

b) ~~A B C D~~
A D C B ✓ (~~C~~ should be always before B)

D A C B ③
~~A C B D~~

A C D B ✓

c) write in pipe line 7, 8
read from pipe line 12, 17

✓ A B C D , A B D C ✓
✓ A D B C A D C B ✓ ③
✓ A C B D
✓ A C D B

d) ~~A C~~
~~A D B~~
~~A B~~
~~A D C~~

A	A	A
D	D	B
B	C	D
C	B	C

③

a)

A	A	A	A	A	A	D	D
B	B	C	C	D	D	A	A
C	D	D	B	C	B	B	C
D	C	B	D	B	C	C	B

③

Scanned with CamScanner

ex3.c

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
#define N 10

void main(){
        int p1[2],p2[2];
        char c='A';
        pipe(p1);
        pipe(p2);
        int i;

        for (i=0;i < N;i++)
          if (!fork()) break;

        while(1) {

                if (i< N && i%2==0) {  // even child
                  sleep(2);
                  write(p1[1], &c, 1);
                }
                else if (i < N && i%2!=0) {
                  sleep(2);
                  read(p1[0],&c,1);
                printf("i'm a child process with odd pid  %d, i read the character %c  and with i=%d\n",getpid(),c,i);
                }

        } // end While

} // End main
```

5

ex3_b.c

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
#define N 10
#define M 20

void main(){
        int p1[2],p2[2],n=0; char c='A';          pipe(p1); pipe(p2);  int i;
        write(p2[1],&n,sizeof(int));

        for (i=0;i < N;i++)
                if (!fork()) break;

        while(1) {

        if (i< N && i%2==0) {  // even childs -- producers

                close(p1[0]);
                printf("I'm producer child with even pid %d, i will write the character %c and with
                i=%d\n",getpid(),c,i);
                read(p2[0],&n,sizeof(int));
                printf("the value of n -- producer -- is %d\n",n);

                 if (n < M) {
                        write(p1[1], &c, 1);
                        n++;
                        write(p2[1],&n,sizeof(int));
                 }

                 Else  {
                        write(p2[1],&n,sizeof(int)); }
                 sleep(1);
                 }
         else if (i < N && i%2!=0) {  // odd childs -- consumers
                 sleep(2);
                 printf("i'm a child process with odd pid  %d, i read the character %c  and with i=%d\n",getpid(),c,i);
                 close(p1[1]);
                 read(p1[0],&c,1);
                 read(p2[0],&n,sizeof(int));
                 printf("the value of n -- consumer -- is %d\n",n);
                 n--;
                 write(p2[1],&n,sizeof(int));
          } // End if

   } // end While

   } // End main
```

7

3