Lebanese University	I3303- E- Operating system II	Second Session 19-20
Faculty of Sciences 1		Duration: 75 minutes

Problem I (17+18) (35 points)

1) We would like to simulate the work of a walkie-talkie between two processes, a father and a son. Each of the two processes reads words from the keyboard simulating typing a message when it reads the keyword "over" it stops and sends a signal to the other process to begin reading. The keyword "out" means that the communication has ended. The process that reads the "out" message stops and should stop his partner.

17

P.S: the father starts the communication

2) What is the possible output of the following pieces of programs

Program1

```
void main()
{
      int i, j = 1, p[2];
      pipe(p);
      write(p[1],&j,sizeof(int));
      for(i=1;i<5;i++)
            if(!fork()){
                  close(p[1]);
                  break;
            }
      read(p[0],&j,sizeof(int));
      printf("%d\n",i);
}
```

6

Program2

```
void main()
{
      int i, j = 1, p[2];
      pipe(p);
      for(i=1;i<5;i++)
            if(!fork()){
                  close(p[1]);
                  break;
            }
      wait(0);
      read(p[0],&j,sizeof(int));
      printf("%d\n",i);
}
```

6

Program3

```
void main()
{
      int i, j = 1, p[2];
      pipe(p);
      for(i=1;i<5;i++)
            if(!fork()){
                  close(p[1]);
                  break;
            }
      write(p[1],&j,sizeof(int));
      printf("%d\n",i);
      read(p[0],&j,sizeof(int));
}
```

6

Lebanese University
Faculty of Sciences 1

I3303- E- Operating system II

Second Session 19-20
Duration: 75 minutes

Problem 2 (19 + 16) (95 points)

I. Consider a system with paginated main memory. The memory is composed of 4 frames (frames) each has a size of 64 bytes. At a given moment, the memory is empty, then two processes P1 (4 pages) and P2 (2 pages) are launched in the system. The processor sends requests submissions in the following order in the format [hexadecimal logical address, Process]:

[3F, P1]
[4A, P1]
[1D, P1]
[00, P2]
[CA, P1]
[87, P1]
[39, P2]
[2B, P1]
[00, P1]
[11, P2]

19

1. What is in bytes the size of the main memory? 2
2. What is the size of the virtual address space? 3
3. Indicate by figures the evolution of the memory and the number of page faults using the following page replacement algorithms:
   a) FIFO 4
   b) LRU 4
   c) Second chance. 4

II. We assume the existence of six free space holes in a contiguous allocation memory, named from A to F, as shown in the right table:

(16)

Hole	Address	Size
A	360	20 Bytes
B	500	12 Bytes
C	1200	18 Bytes
D	1400	4 Bytes
E	1820	40 Bytes
F	1960	32 Bytes

Three successive R1, R2 and R3 requests of size 12, 10 and 6 bytes must be satisfied. What is the memory address allocated to each of these requests if the allocation strategy is:

a. First Fit ? 4
b. Best Fit ? 4
c. Worst Fit ? 4
d. Next Fit ? 4

2) Prog1 : (6 pts)

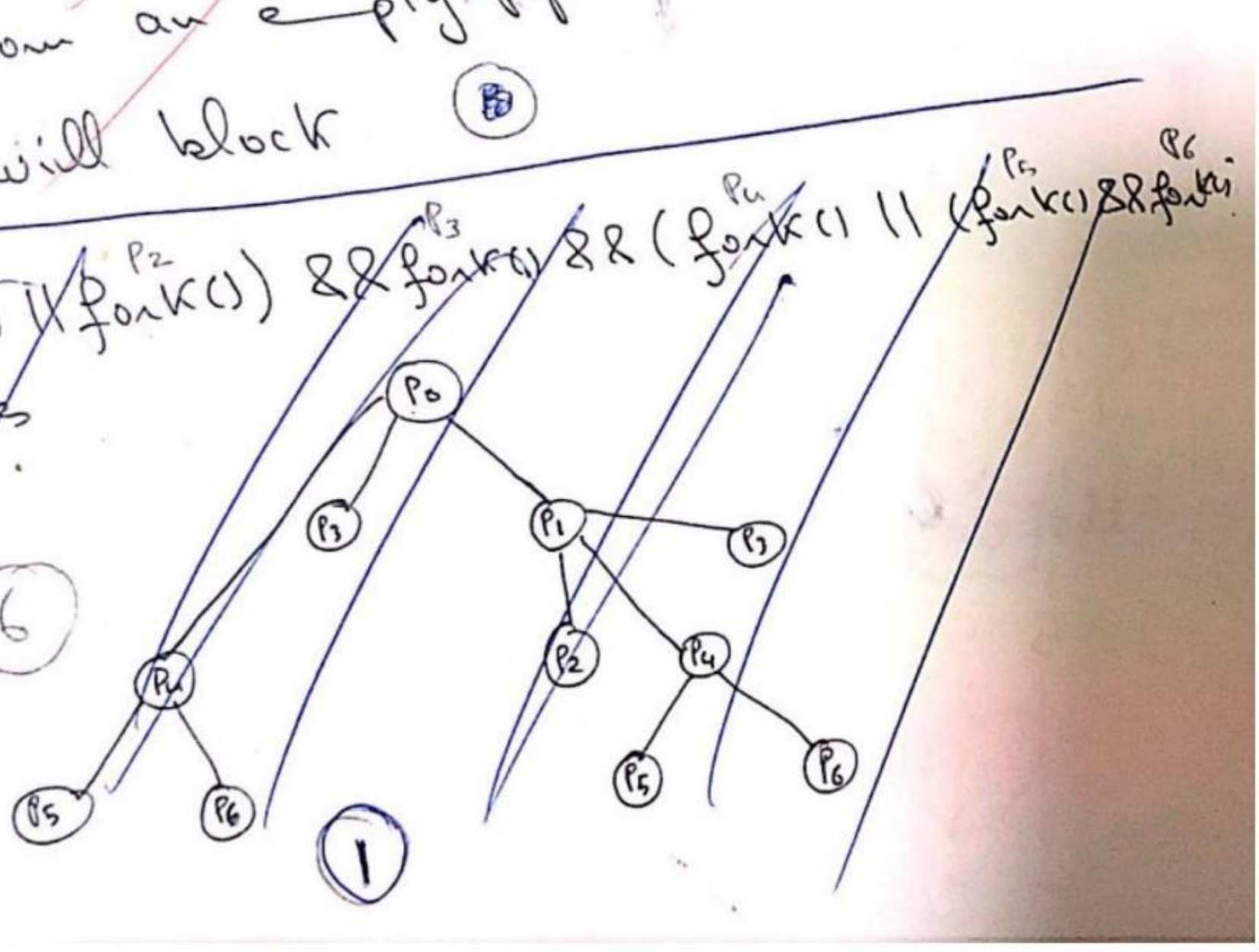
> if the parent reads first, then all child will print also 5 4 3 2 1 (0) (because the parent exits and the write side on childs is closed) and exit

> if any child read first, it prints the value of i (ex. 4)
all other processes included parent will block because they will attempt to read from an empty pipe with write side opened. (0)

Prog2 : (6 pts) No output

here the parent cann't read immediately, he must waiting at least one child to read from pipe and exit.
Suppose the fourth child read from pipe, it will print 4 and exit.
Now the parent or other child will attept to read from an empty pipe with write side opened
So they will block (0)

B) (fork() || fork()) && fork() && (fork() || (fork() && fork()))
P1 P2 P3 P4 P5 P6
14 processes



Prog 3 (6 pts)

- each child reads from empty pipe with no writer (closed port) => no block
- then it ~~writes~~ prints the value of i

possible outputs :

```
1 2 3 4 | 5
2 1 3 4 | 5
⋮       |
        | 5
```

Possible combination of {1, 2, 3, 4}

- the parent prints 5 at the end after the ending of childs

- ~~after the process~~ child

if the parent starts first, it can write into the pipe the value 5, prints 5, and read 5, ~~then exit~~

- all other child will print its value of i then they will block on reading the empty block if the parent doesn't exist yet. If not they will not block

if parent first

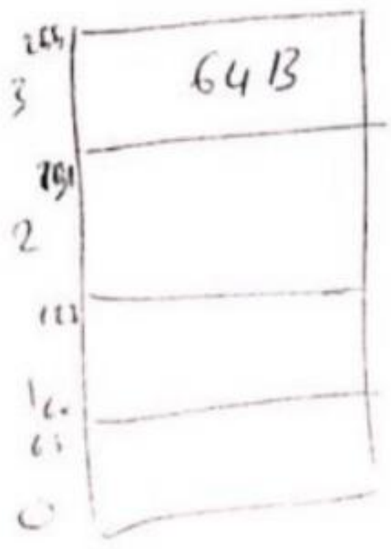
```
+ 5 5 ... 5
  1 2     4
  2 1     3
  3 3     2
  4 4 ... 1
```

if any of the child starts first

suppose child nb 2

```
2 2 2 2
1 3 4 5
3 1 3 1
5 5 5 1
```

2: Memory management (15 pts)



$P_1 \rightarrow$ 4 pages

$P_2 \rightarrow$ 2 pages

1) size of array:

$64 \times 4 = 256$ Byte $= 2^8$ Byte (2)

2) size of the virtual address space

Page size $= 64$ By $= 2^6$ Bytes (2)

$\Rightarrow$ 6 bits is needed for the offset

- The logical address is in hexadecimal (2 digits)
- each is encoded on 4 bits $\Rightarrow$ the logical address is encoded on 8 bits $\Rightarrow$ # (2) | offset (6)

(virtual address on 8 bits)

$\Rightarrow 2^2$ pages $\Rightarrow 4 \times 64 = 256$ Bytes $= 2^8$ (3)

3)

		Page	
3F $\Rightarrow$	0011 1111	63	0	P_1
4A $\Rightarrow$	0100 1010	74	1	P_1
1D $\Rightarrow$	0001 1101	29	0	P_1
00 $\Rightarrow$	0000 0000	0	0	P_2
CA $\Rightarrow$	1100 1010	202	3	P_1
87 $\Rightarrow$	1000 0111	135	2	P_1
39 $\Rightarrow$	0011 1001	57	0	P_2
2B	0010 1011	43	0	P_1
00	0000 0000	00	0	P_1
11	0001 0001	17	0	P_2

(3)

$\Rightarrow$ ~~He~~ $W = \{0, 1, 0, 0, 3, 2, 0, 0, 0, 0\}$

$W = \{(0;P_1), (1;P_1), (0;P_1), (0;P_2), (3;P_1), (2;P_1), (0;P_2), (0;P_1), (0;P_1), (0;P_2)\}$

4 ~~page faults~~

a) FIFO

0	0	0	0*	0*	0	0	0*	0*	0*
1		1	1		1	1	1	1	1
2					3	3	3	3	3
3						2	2	2	2

(2)

(8)

FIFO

Page refs	Fault?	4 page frames: Page contents			
(0, P1)	✓	(0,P1)			
(1, P1)	✓	(1,P1)	(0,P1)		
(0, P1)	X	(1,P1)	(0,P1)		
(0, P2)	✓	(0,P2)	(1,P1)	(0,P1)	
(3, P1)	✓	(3,P1)	(0,P2)	(1,P1)	(0,P1)
(2, P1)	✓	(2,P1)	(3,P1)	(0,P2)	(1,P1)
(0, P2)	X	(2,P1)	(3,P1)	(0,P2)	(1,P1)
(0, P1)	✓	(0,P1)	(2,P1)	(3,P1)	(0,P2)
(0, P1)	X	—	—	—	—
(0, P2)	X	—	—	—	—

6 page faults (3) (4)

b) LRU

Page refs	Fault	4 Page Frames: Page contents			
(0, P1)	✓	(0,P1)			
(1, P1)	✓	(1,P1)	(0,P1)		
(0, P1)	X	(0,P1)	(1,P1)		
(0, P2)	✓	(0,P2)	(0,P1)	(1,P1)	
(3, P1)	✓	(3,P1)	(0,P2)	(0,P1)	(1,P1)
(2, P1)	✓	(2,P1)	(3,P1)	(0,P2)	(0,P1)
(0, P2)	X	(0,P2)	(2,P1)	(3,P1)	(0,P1)
(0, P1)	X	(0,P1)	(0,P2)	(2,P1)	(3,P1)
(0, P1)	X	—	—	—	—
(0, P2)	X	(0,P2)	(0,P1)	(2,P1)	(3,P1)

5 page faults (3) (4)

c) Second chance

	Page refs	Fault?	4 Page Frames: Page contents				out
A	(0, P1)	✓	(0,P1)				
B	(1, P1)	✓	(1,P1)	(0,P1)			
A	(0, P1)	X	(1,P1)	(0,P1)*			
C	(0, P2)	✓	(0,P2)	(1,P1)	(0,P1)*		
D	(3, P1)	✓	(3,P1)	(0,P2)	(1,P1)	(0,P1)*	
E	(2, P1)	✓	(2,P1)	(3,P1)	(0,P2)	(0,P1)	(1,P1)
C	(0, P2)	X	(2,P1)	(3,P1)	(0,P2)*	(0,P1)	
A	(0, P1)	X	(2,P1)	(3,P1)	(0,P2)*	(0,P1)*	
A	(0, P1)	X	—	—	—	—	
C	(0, P2)	X	—	—	—	—	

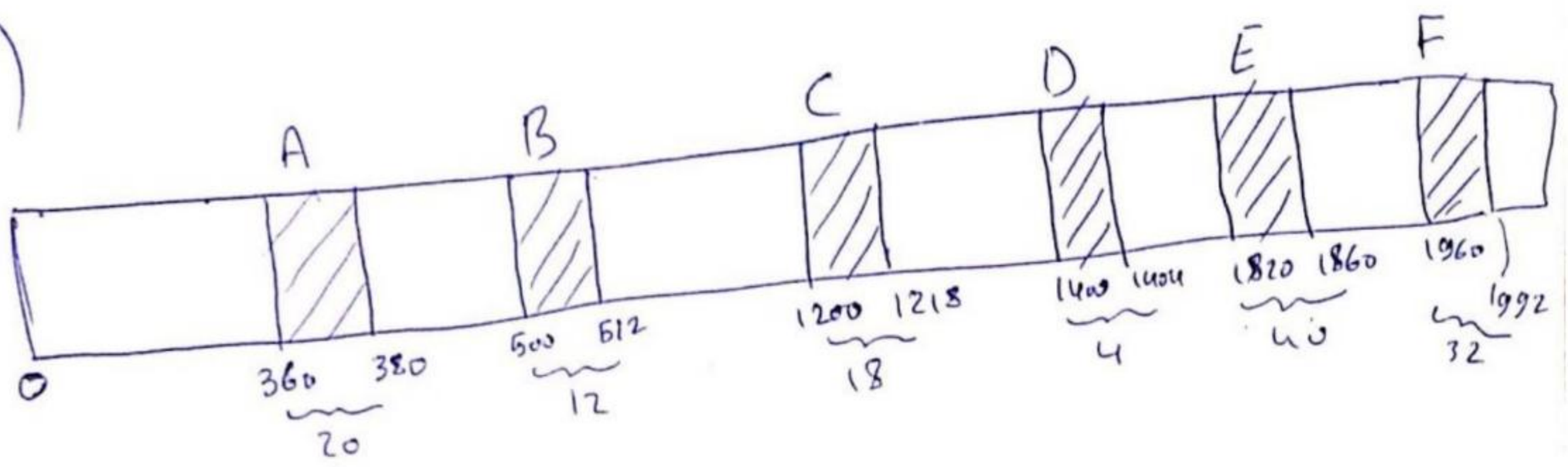
~~5~~ page faults

6 page faults

(3) (4)

Scanned with CamScanner

I3303 - E 2nd 19-20



R_1 : 12 bytes
R_2 : 10 bytes
R_3 : 6 byte

a) First Fit

R_1 is allocated on A ⇒ A = 8 bytes
R_2 is allocated on B ⇒ B = 2 bytes
R_3 is allocated on A ⇒ A = 2 bytes

(4)

b) Best Fit

R_1 is allocated on B ⇒ B = 0
R_2 is allocated on C ⇒ C = 8
R_3 is allocated on C ⇒ C = 2

(4)

c) worst Fit

R_1 is allocated on E ⇒ E = 28
R_2 is allocated on F ⇒ F = 22
R_3 is allocated on E ⇒ E = 22

(4)

d) Next Fit

(4)

- R_1 on A ⇒ A = 8
- R_2 on B ⇒ B = 2
- R_3 on C ⇒ C = 12

(4)

Problem I 2nd session 19-20

(17 pts)

File: /home/parallels/Documents/FS1...ms/19-20/Second Sessio/Pb1-1.c Page 1 of 1

```
void handler(){}   2

void main(){
signal(SIGSTP handler);  1
signal(SIG CONT handler);  1
char word[  ];

if(!pid=fork()){  1
 pause();         1
1 while( ){       1
      scanf(  , name);  1
      if !(strcmp(name, over )){
   1     kill(getppid(),SIG CONT);
         pause();
      }
      elseif  !(strcmp(name, "out" )){
   1     kill(getppid(),SIGQuit);
         exit( );  1
      }
  }
}
else {            1
    while( ){
      scanf(    , name);  1
      if !(strcmp(name, over )){
         kill(pid,SIG CONT);  1
         pause();          1
      }
      elseif  !(strcmp(name, "out" )){
         kill(pid,SIGQuit);  1
         exit( );
      }                    1
   }

}
}
```