| Lebanese University<br>Faculty of Sciences I | I3303 | Final 2021-2022<br>Duration: 90 minutes |
|---|---|---|

## Problem I — Process — 35 Points

**Part A:** Draw the tree generated by the following code:

5

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

**Part B :** Consider the following program:

```
const int n=3, m=2;
int main(){
      int i, j=0;
      for (i=0;i<n && j<m; i++){
            if ( !fork()){
                  i=0; j++;
            }
      }
printf("j=%d\n", j);
exit (0);
}
```

7 1. Draw the tree of processes generated by the above code. What is the value of **j** displayed by each of the processes? What does it correspond to?

3 2. Complete the previous code so that the processes without children run an executable "toto" which is in the current directory.

3. Complete the code obtained in (2) so that:

10
   a. Each parent process retrieves the total number of child processes created by all its descendants and displays it on the screen.
   b. The main process displays on the screen the total number of child processes created by it and all its descendants.

4. Complete the code in (3) so that:

10
   a. All processes without children redirect their standard output to an anonymous pipe before executing "toto".
   b. The main process retrieves and displays on the screen all the messages written by the processes without children in the pipe, just after the end of all its children.

## Problem II — Memory management — 45 Points

(The two parts are independent)

**Part A:** (20)

We consider a 2-level paging system in which the addresses (virtual and physical) are coded on 32 bits. The maximum size of each page table (regardless of its level) is 64 KB. Each entry in a page table consists of 64 bytes.

5 1. Give the format of a virtual address. What is the size of a page?

2. Give the physical address of virtual address 0x 0010 1210, if the page referenced by this address is loaded into memory frame 5 (numbering starts at 0). Give the page number referenced by this virtual address. 0x 0010 1210 (5)
3. Assume a process that references pages 244,257,150,256,257,244,256,120,257 in order. The system reserves the first 3 assumed empty memory frames for the process and uses local allocation. Give the number of page faults generated by the process for each of the following page replacement algorithms: (10)
   - LRU and Optimal.

(25)

**Part B:**

Consider a 2-level paging system in which virtual addresses are encoded in 32 bits and the size of a page is 4KB. The entry for each page table, regardless of level, is 8 bytes.

(16)

1. What is the maximum size in number of pages of a process's address space?
2. Give the number of pages there are in a virtual address space of 4MB.
3. Give the page number that corresponds to the virtual address 0x00110A10.
4. Assume that a process's page tables are in physical memory while the process is running. Give for each of the following configurations, the total size (in number of pages) of the page tables to be kept in memory during the execution of a process of 4 MB of contiguous virtual address space: (10)
   a. The first level page table has 1024 entries.
   b. The first level page table has 2048 entries.
   c. The first level page table has 512 entries. It is for which configuration a, b or c, do you obtain the minimum size?

| Problem III | File System | 20 points |
|---|---|---|

We propose to study a sequential file system management where files reside on a single disk. Descriptors files are opened in main memory in the following table fdesc:

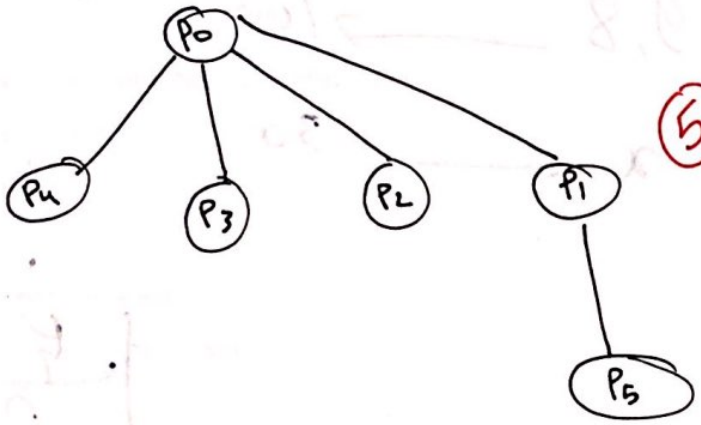
```
#define maxf
struct{
        int lg;
        int topo[8];
        int map1[1024];
        int map2[1024];
        char buffer[2048];
        ...
} fdesc[maxf];
```

The 8 fields in the structure topo contain each the number of map block of level1 that contains each 1024 number of map blocks (level 2). Each block of level 2 may contain 1024 number of data blocks.

1. What is the size of a block in this system? justify (5)
2. How many bytes the number of a block occupies? justify (5)
3. What is the maximum number of blocks of maps (first and second level)? Justify. (5)
4. What is the maximum size (in bytes) of a file supported by this FS? Justify. (5)

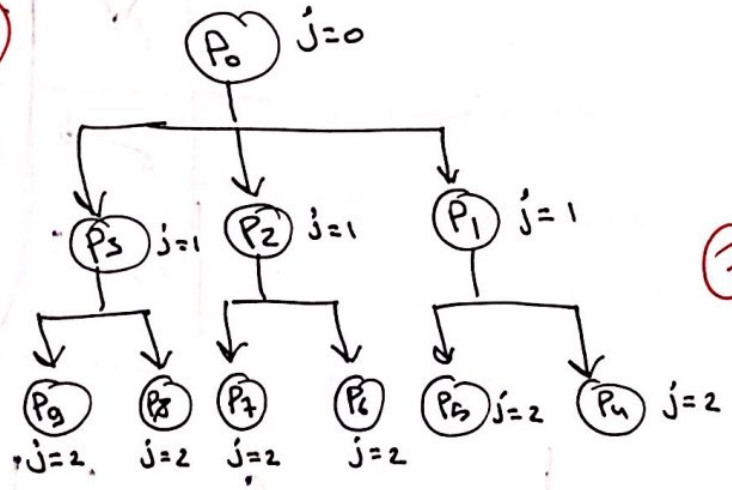
Problem I : 6

Part A : (5)




5

Part B : (30)




1) 7

2) if (j==2)
execl ("toto", "toto", NULL);

3)

- tree → 7 { 4 (tree) + 3 (display) }
- B-2 → 3
- B-3 (a&b) →
- redirection → 9 B-4 (a&b)

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <wait.h>

const int n=3, m=2;
void main(){
  int i, j=0;
  int nbFirst=0, nbSecond=0, nbTotal=0, status, fd[2];
  pipe(fd);
  for (i=0;i<n && j<m; i++){
     if(!fork()){
         i=0;
         j++;
     }
     else {
       if(j==0){  // main process
          nbFirst++;}
       else if (j==1){  // first level
          nbSecond++;}
     }
   }  // End For

    if(j==0){  // main process
      while(wait(&status)!=-1)
        nbTotal +=WEXITSTATUS(status);
      printf("Main process with pid %d and i created %d descendants\n", getpid(),
 nbTotal+nbFirst);
      dup2(fd[0],0);
      while(read(fd[0], &messg, strlen(messg)+1)
         printf("message written by child level 2 is:%s/n",messg);
    }
    if(j==1){ // First level
      printf("Child with pid=%d, i created childs=%d\n",getpid(),nbSecond);
      exit(nbSecond);
    }
    if(j==2) { // processes without child (Second level)
      dup2(fd[1],1);
      execl("toto","toto",NULL);
    }
 }
```

## Problem II

### Part A

* 2-level paging system
* 32 bits addressing
* size (page table) = 64 KB
* size (PTE) = 64 B

1) Format of virtual address?

virtual address : level 1 | level 2 | offset

Nb of ~~pages~~ entries $= \frac{64\,KB}{64\,B} = 2^{10}$ ~~pages~~ entries

$\Rightarrow$ we have 10 bits for each level

~~for the offset~~

5

10 | 12

10

page size? : page size $= 2^{12}$ $\boxed{= 4KB}$

2) 0X0010 1210 = (0000 0000 0001 0000 0001 0010 0001 0000)$_2$

Pg#1 Pg#2 offset

virtual page = 0 loaded in physical frame 5

offset = 528 (level 1 → 0, level 2 → 257)

$\Rightarrow$ physical address = 5 × (page size) + offset

$= (5 \times 4096) + 528$

$\boxed{= 21008}$

5

3

3) w = {244, 257, 150, 256, 257, 244, 256, 120, 257}
A B C D B A D E B

LRU

| Pages | Fault | Memory F1 | F2 | F3 |
|---|---|---|---|---|
| A | y | A | | |
| B | y | B | A | |
| C | y | C | B | A |
| D | y | D | C | B |
| B | N | B | D | C |
| A | y | A | B | D |
| D | N | D | A | B |
| E | y | E | D | A |
| B | y | B | E | D |

7 page faults

5

Optimal

| Pages | Fault | F1 | F2 | F3 |
|---|---|---|---|---|
| A | y | A | | |
| B | y | B | A | |
| C | y | C | B | A |
| D | y | D | B | A |
| B | N | D | B | A |
| A | N | D | B | A |
| D | N | D | B | A |
| E | y | D | B | E |
| B | N | D | B | E |
| | | | | |

5 page faults

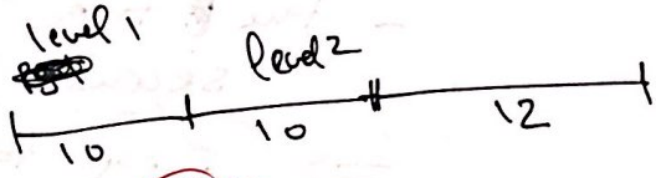
5

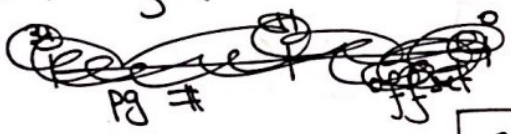
Scanned with CamScanner

## Problem II (Part B)

* 2 level paging system
* 32 bits addressing
* page size = 4 KB = $2^{12}$
* PTE = 8 Bytes

1) Max size?




The max size is : $\boxed{2^{20} \text{ pages}}$ 5

2) 4MB address space

$4MB = 2^2 \times 2^{20}$

Nb of pages $= \frac{2^{22}}{2^{12}} = \boxed{2^{10} \text{ pages}}$ 5

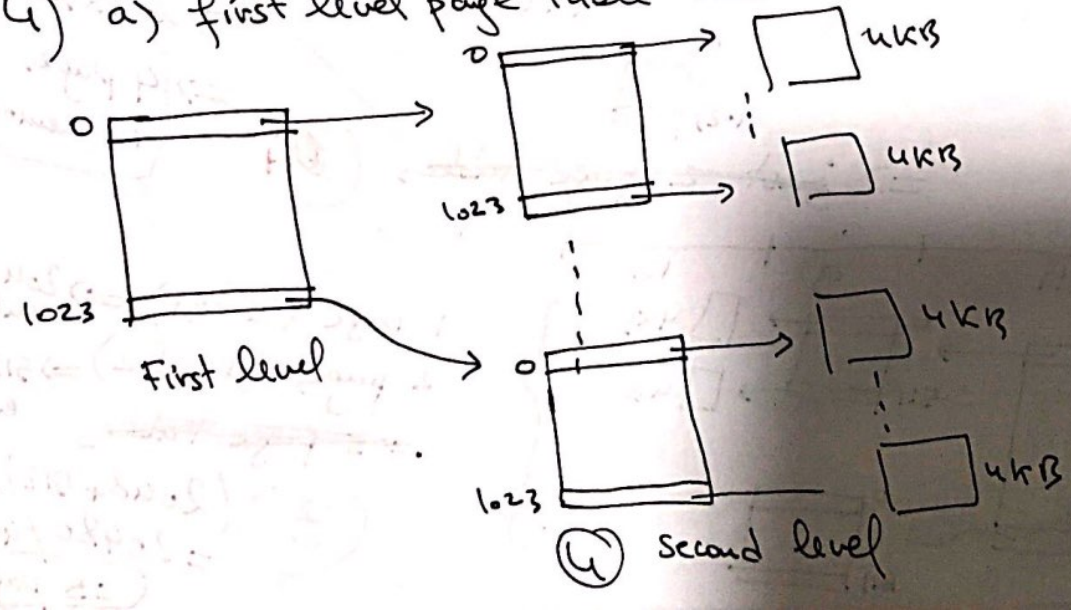
3) 0X 00110A10

$(\underbrace{0000\ 0000\ 0001\ 0001\ 0000}_{\text{pg \#}} \mid \underbrace{1010\ 0001\ 0000}_{\text{offset}})_2$

272 2576

virtual page = $\boxed{272}$ 5

4) a) first level page table has 1024 entries




4

~~Each page in the second level has:~~

$$\cancel{\frac{4KB}{8B} = \frac{2^{12}}{2^3} = 2^9 \text{ entries} = 512 \text{ entries}}$$

~~For a contiguous process with 4MB size we need:~~

~~- first level page table~~

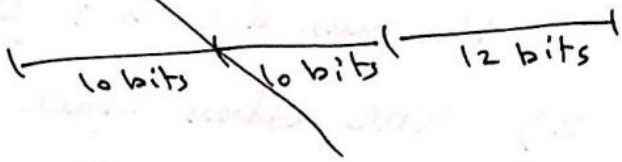
~~- the two first page tables of the second level~~

~~$\Rightarrow$ 3 page tables 1 + 2~~

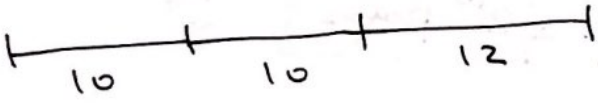
~~b) 2048 entries~~

~~the same as a~~

~~c) 512 entries~~

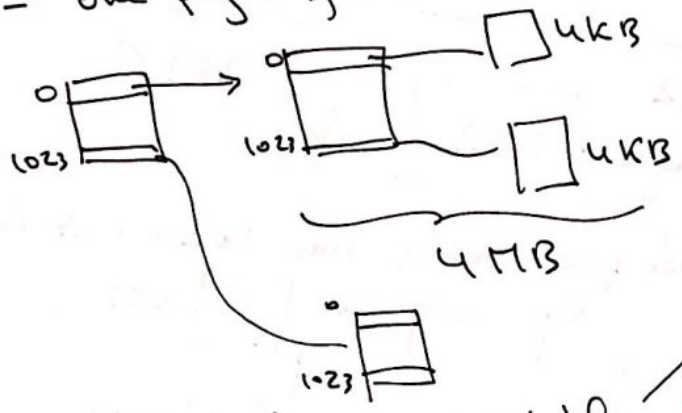



a)




we need:

- the first level page table
- one page from the second level



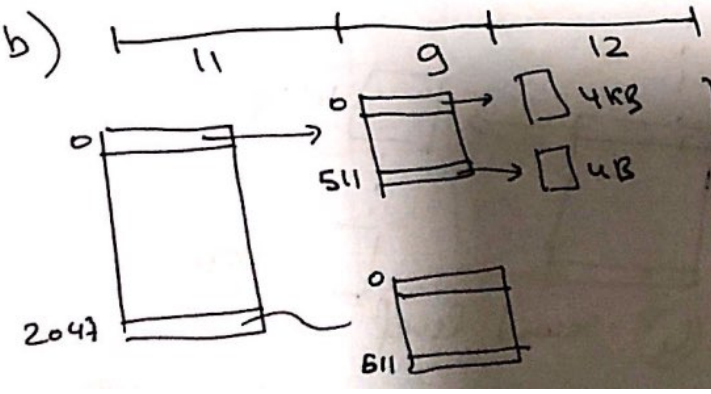

- First level page table
+ second level "
= 2048 entries
- each entry = 8 bytes

$\Rightarrow 2048 \times 8$

$\Rightarrow$ 4 pages in memory

$\Rightarrow$ ~~just 2 page tables~~ 4

b)




1 page (level 1) $\Rightarrow$ 2048 entries

2 pages (level 2) $\Rightarrow$ 512 entries

~~= 3 page table~~

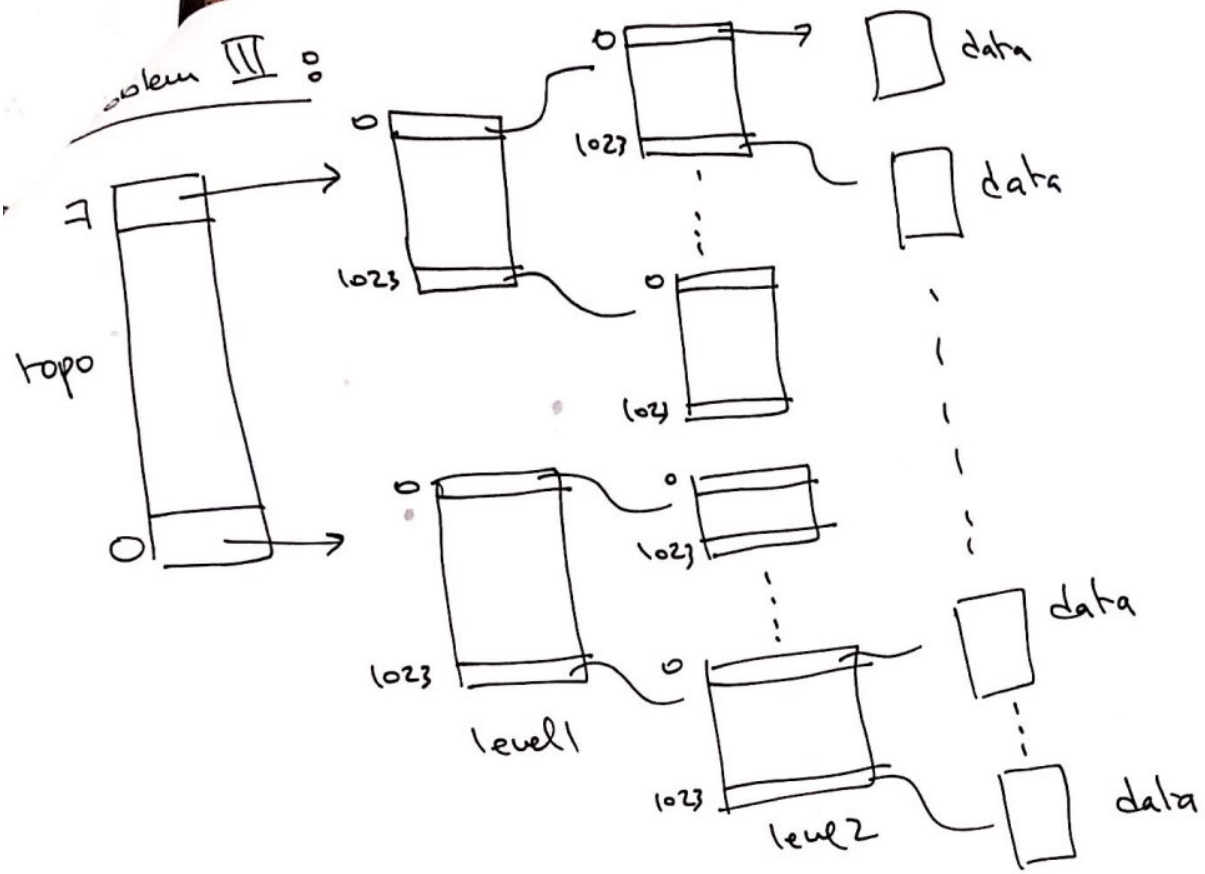
3

$(2048 + 512) \times 8$

$= 20480 / 4096$

$= 5$ pages

Problem II :




1) size of a block?

5 size = 2048 bytes (size of the buffer)

2) Each map block has 1024 entries

so each entry occupies $\frac{2048}{1024} = 2$ bytes

5 => $\boxed{\text{the block \# is 2 bytes}}$ 5

3) Max nb of map blocks is : $8 + (8 \times 1024)$ map blocks

4) Max size of a file : $8 \times (1024)^2 \times 2048$ bytes 5

---

Contine Pb II (Part B) 3

$(512 + 2048) \times 8 = 20480$

$20480 / 4096 = \boxed{5 \text{ pages}}$

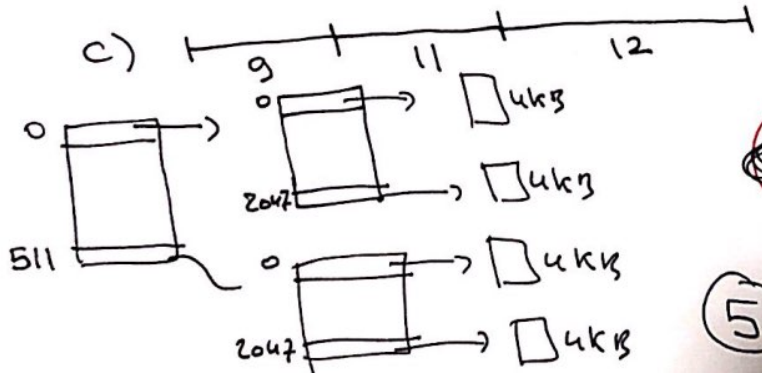
we need :

- first level page table

5 and

- one page table from level 2

c)




5 => ~~2 page tables~~