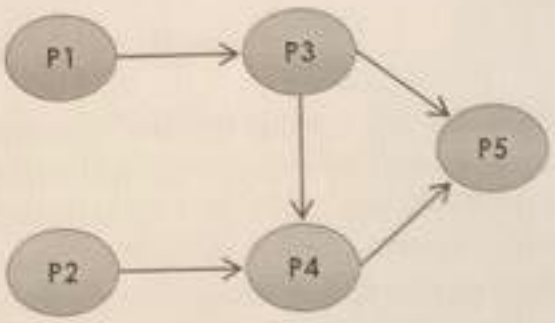
Lebanese University Faculty of Sciences I	INFO324 - E	Final 2016-2017 Duration 2 hours

18

Problem I: Process management (20 points):

1. Write a program C under UNIX where a parent process creates 5 processes P1, P2, P3, P4 and P5. Assume that each Pi process executes the Fi () function (you are not required to write the code). Moreover, using the signals, the execution order of the processes must respect the order presented by the following graph (for example, according to the graph P3 must not be executed before P1):

12



2. Taking into account the solution of the preceding question, and assuming that the codes of the functions Fi () are the following:

6

F1() { i = i-5; }
F2() { i = i*3; }
F3() { i = i-8; }
F4() { i = i*3; }
F5() { i = i+3; }

(i is considered as variable common to all processes and initialized to the value 5 at beginning), provide all possible values of i after the execution of the five processes.

22

Problem II: Memory management (20 points):

A) Consider the following reference set to the following virtual pages:
3, 4, 3, 2, 1, 3, 5, 1, 4, 3, 1, 3, 5, 2, and 3
In a memory system with 3 frames. How many page faults is generated for each of the following replacement algorithms:

10 5×2

a. LRU
b. Second chance.

B) Below is given a part of the table of segments of a process:

Segment #	Size of segment	Base address
1	30 KB	.....
2	16 KB	32 KB
3	....	105 KB
4	8 KB	58 KB

6

a) Give the physical address of the following virtual address (2, 5703) 2
b) the physical address of a data in segment 4 is 67502 bytes, find its virtual address 2
c) complete the missing values in the above table knowing that the virtual address (1, 2453) corresponds to physical address 75157 and the physical address of the last data in segment 3 is 128000 2

C) We consider a paginated memory system in two levels with pages of size 4kb each. the addressing scheme is byte by byte on 32 bits, where the addressing on the second level is on 10 bits

a. How many hyper-pages (first level) are in the system?
b. Give the decimal address of the following address (39, 1002, 2931).
c. Determine the following address 561 732 092 in the form of (hyper-page, page, offset)

(6)

Problem III: File System (30 points)

A) Consider the following track requests in the disk queue: 95, 180, 34, 119, 11, 123, 62, and 64. Consider that the read/write head is positioned at location 50. Compute the total head movement of the head for a 200 track disk (0-199) according to the following strategies: look and C-Scan

(8)

B) Consider a UNIX file system where the inode structure has 10 direct pointers, one simple indirect, one double indirection and one triple indirection. The size of the block is 1 KB and each block address occupies 4 bytes and the inode size is 32 bytes. In this system, a process that reads sequentially a file with size 8MB stored on disk at 256 bytes at a time makes 32768 read requests. Knowing that the system doesn't have a cache buffer, calculate the number of disk I/O access needed.

(6)

C) Given a FS (file management system) where the topo table contains 8 corresponding entries where: The first 4 entries each correspond to a single level of indexing (each pointing to a map block). The last 4 entries point directly to data blocks.
Given that the size of each block is 1 kilobytes, the number of a block occupies 2 bytes and that each inode block contains 32 inodes:

(9)

1. What is the maximum size of a file supported by this FS?
2. What is the effective space occupied on the disk for the file of maximum size?
3. Consider a file of 1,500,000 bytes. How many blocks (data and maps) are needed to represent this file on disk? Briefly justify your answer.

D) In a UNIX file system, we want to open the following file: /usr/ast/mbox. Given that the inode and data blocks of the root folder is loaded into memory. Describe in details the steps to do for opening the file (on making the needed assumptions) and calculate the number of disk I/O requests.

(7)

P_1, P_2, P_3, P_4, P_5

From	i
P_1	0
P_2	15
P_3	-3
P_4	15
P_5	8

½

P_1	0
P_3	-3
P_2	15
P_4	15
P_5	8

½

P_1	0
P_2	15
P_4	15
P_3	-3
P_5	8

½

P_1	0
P_2	15
P_4	15
P_5	8
P_3	-3

½

P_1	6
P_3	-3
P_5	8
P_2	15
P_4	15

½

P_2	15
P_4	15
P_5	8
P_1	0
P_3	-3

½

15 P_2
0 P_1
15 P_4
-3 P_3
8 P_5

15 P_2
0 P_1
-3 P_3
15 P_4
8 P_5

15 P_2
0 P_1
15 P_4
-3 P_3
8 P_5

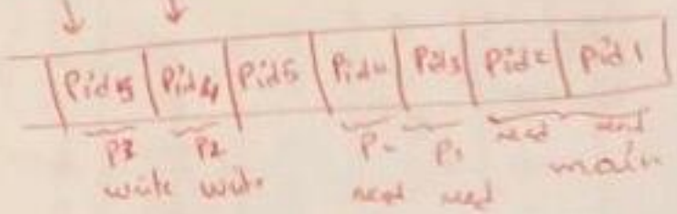
4

Problem 1

C:\Users\Dell\AppData\Local\Temp\ex1-9.c — Tuesday, June 20, 2017 7:09 PM

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <signal.h>
#include <stdlib.h>

int pid, i, fd[2];

void handler(int nsig){
printf("Process id Activated\n", getpid());
//signal(SIGUSR1, handler);
}

void main()
{
 pipe(fd);
 signal(SIGUSR1, handler);
 for(i=1;i<=5;i++) {
   if (pid=fork()) {write(fd[ ], &pid, sizeof(int));}

else
{
switch(i) {
 case 1: pause();
   printf("Process id Activated\n",getpid());
   read(fd[ ], &pid, sizeof(int));
   printf("pid=%d\n",pid);
   kill(pid, SIGUSR1);
   exit(1);
 case 2: pause();
   printf("Process id Activated\n");
   read(fd[ ], &pid, sizeof(int));
   kill(pid, SIGUSR1);
   write(fd[ ], &pid, sizeof(int));
   exit(1);
 case 3:
   pause();
   printf("Process id Activated\n");
   read(fd[ ], &pid, sizeof(int));
   kill(pid, SIGUSR1);
   write(fd[ ], &pid, sizeof(int));
   read(fd[ ], &pid, sizeof(int));
   kill(pid, SIGUSR1);
   exit(1);
 case 4:
   pause();
   pause();
   read(fd[ ], &pid, sizeof(int));
   kill(pid, SIGUSR1);
   exit(1);
 case 5:
   pause();
   pause();
   exit(1);

} // end switch
} // end else
} // end for

if (i==6) {
   read(fd[ ], &pid, sizeof(int)); kill(pid, SIGUSR1);
   read(fd[ ], &pid, sizeof(int)); kill(pid, SIGUSR1);
   while(wait(NULL));
}
```

handler → 1
signal → 1
pipe + fork → 1
switch → 1
i=1 → 1
i=2 → 1
i=3 → 2
i=4 → 2
i=5 → 1
i=6 → 1 (main)



14

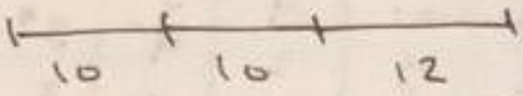
} // end main

21

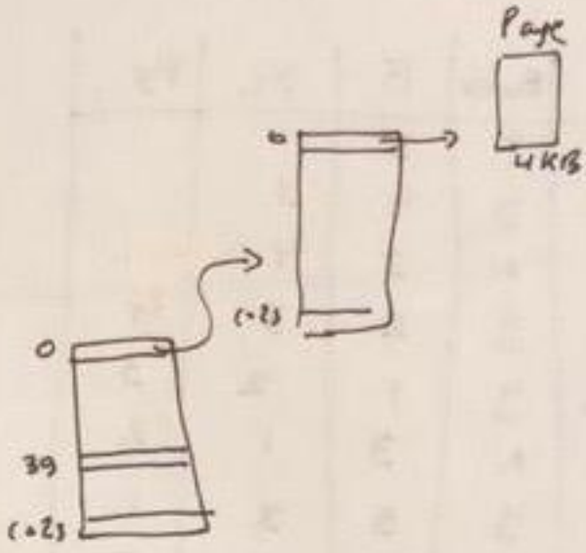
c) 2 levels memory paginated system

- page size = 4KB
- 32 bits address
- second level 10 bits

a) Nb of hyper-pages?



- 2^{10} hyper pages (1)



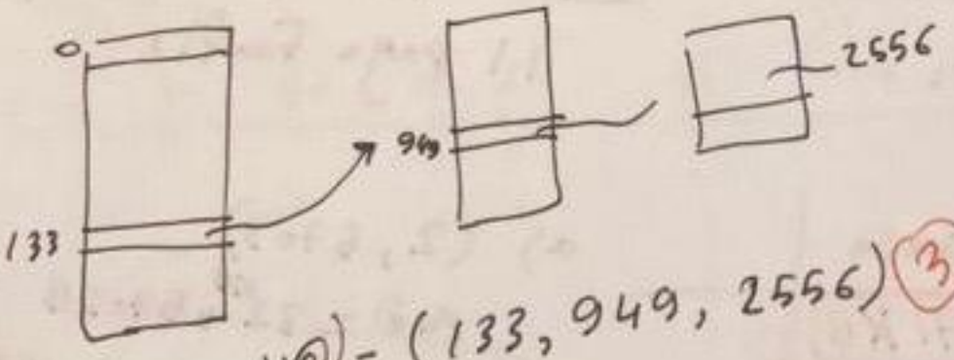
b) (39, 1002, 2931)

each hyper page covers $1024 \times 4KB = 4MB$

phys @ = $39 \times 4MB + 1002 \times 4KB + 2931$ (2)

c) $561\,732\,092 = 133 \times 4MB + 3889660$

$3889660 = 949 \times 4KB + 2556$



V@ = (133, 949, 2556) (3)

INFO 324-E 2016-2017

Solution

Pb II

A)

3	Fault	F1	F2	F3
3	✓	3		
4	✓	4	3	
3	✓	3	4	
2	✓	2	3	4
1	✓	1	2	3
3	✓	3	1	2
5	✓	5	3	1
1	✓	1	5	3
4	✓	4	1	5
3	✓	3	4	1
1	✓	1	3	4
3	✓	3	1	3
5	✓	5	3	1
2	✓	2	5	3
3	✓	3	2	5

9 page Faults (5)

3	Fault	F1	F2	F3
3	✓	3		
4	✓	4	3	
3	✓	4	3*	
2	✓	2	4	3*
1	✓	1	2	3
3	✓	3	1	3*
5	✓	5	3*	3
1	✓	1	5	1
4	✓	4	1*	5
3	✓	3	4	1
1	✓	1	3*	4
3	✓	3	1	3
5	✓	5	5	1
2	✓	2	2	5
3	✓	3	3	5

11 page Faults (5)

B)

Seg #	size	Base
1	30KB	71KB (1)
2	16KB	32KB
3 (1)	20KB	105KB
4	8KB	58KB

a) (2, 5703)

phy @ = 32^KB + 5703 B

= (32 × 1024) + 5703

= 38471 bytes (2)

b) phy @ = 67502 (seg 4)

virtual @ = (4, 8110) (2)

67502 = 65 × 1024 + 942

67502 − (58 × 1024) = 8110

c) (1, 2453) → 75157

seg 3 (last data = 128000)

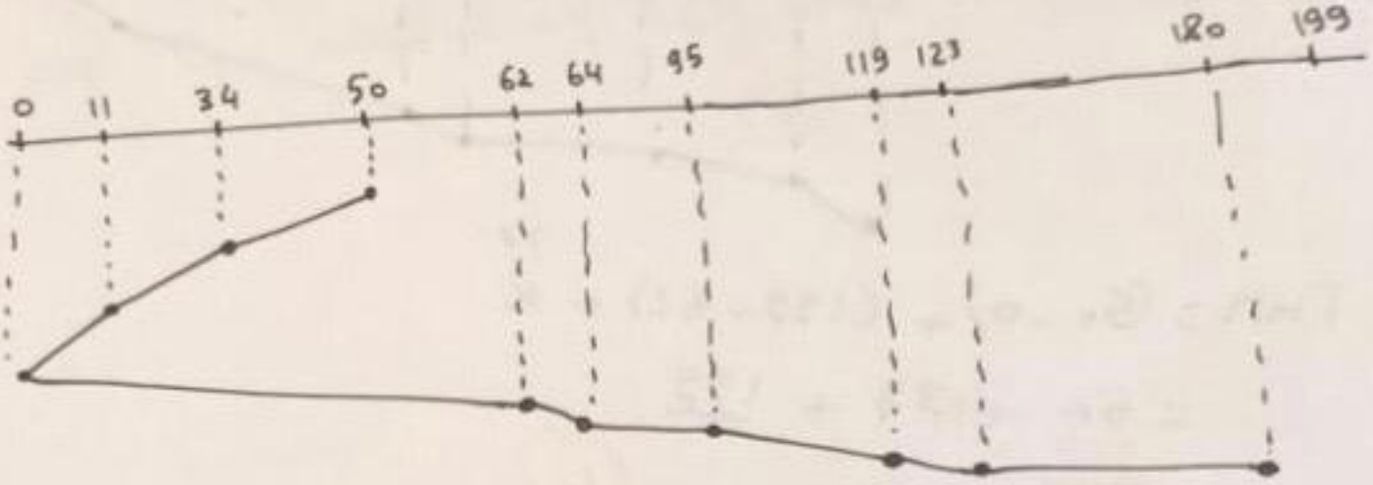
128000 = 125 KB

(2)

(Continue)

Problem III (continue)

A) Look and C-Scan

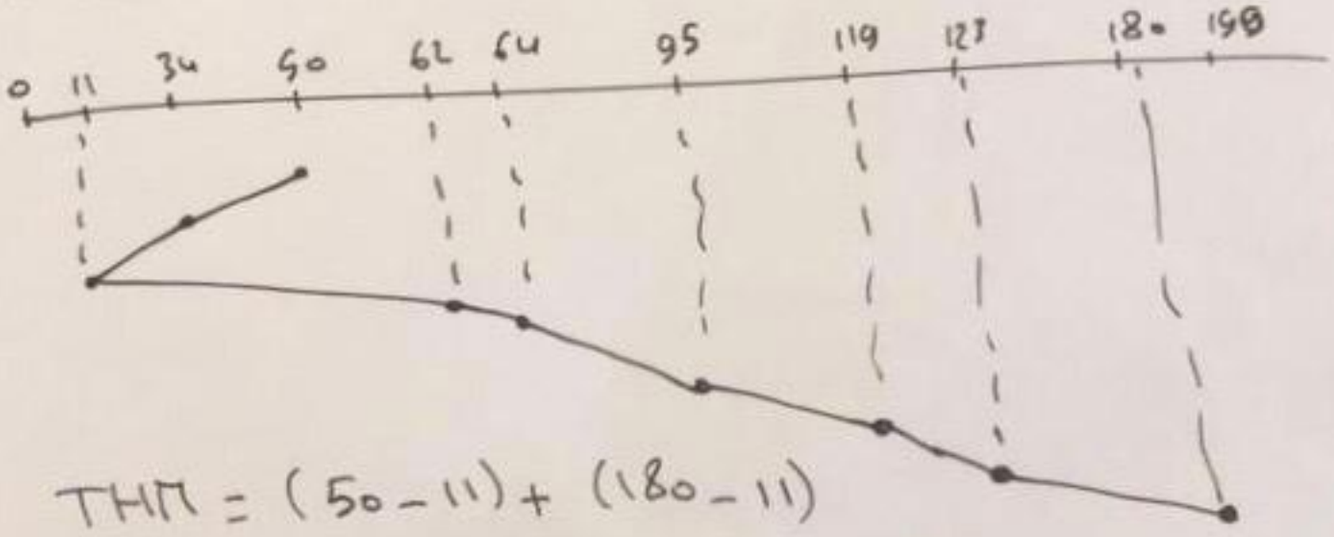


THM = (50 - 0) + (180 - 0)

= 50 + 180

= 230 (Scan)

Look:



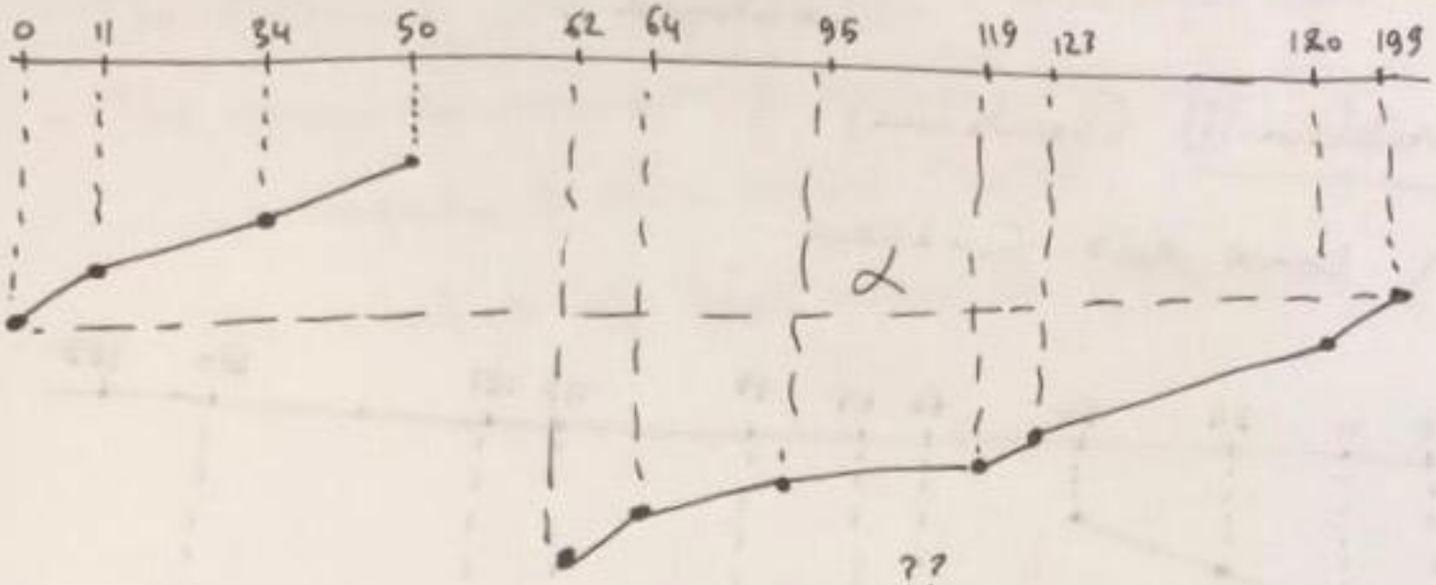
THM = (50 - 11) + (180 - 11)

= 39 + 169

= 208 tracks

(4)

C-Scan



$THM = (50 - 0) + (199 - 62) + \alpha$??

$= 50 + 137 + 199$

$= 386$

(4)

Problem II (continue) (7)

D) /usr/ast/mbox

- the inode of the root is loaded into memory
- also the data blocks of the root are loaded so:

① read the content of the root folder (no disk access)

suppose it is as follows:

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

⇒ the inode number of the folder usr is 6

② read the block that contains the inode # 6 (1 I/O access)

③ the inode says that the content of the folder usr is in block 132 (for example)

④ read the block 132 and load it into memory (1 I/O access)

⑤ the content of the folder usr is:

6	.
1	..
26	ast

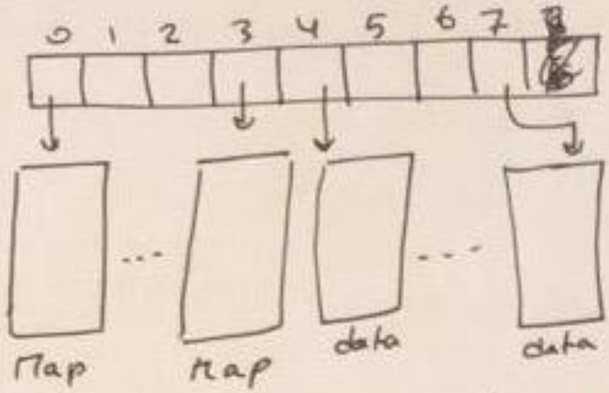
so the inode of the folder ast is 26

⑥ read from disk the block that contains inode 26 (1 I/O access)

(1.5)

Pb III (File System)

c)



- block size = 1 KB = 2^{10}
- block # = 2 Bytes
- block contains 32 inodes

1) Maximum File size ?

each map block contains $\frac{2^{10}}{2} = 2^9$ entries = 512 entries

$$\text{File size (max)} = \underbrace{(4 \times 512 \times 1\,KB)}_{\text{map} \to \text{data}} + \underbrace{4 \times 1\,KB}_{\text{Data}} \quad (2)$$

$$= 2048 + 4 = 2052\,KB$$

2) effective size (disk) = size(data) + size(map) + size(inode)

$$= 2052^{KB} + 4\,(\text{map})^{KB} + 32 \text{ bytes} =$$

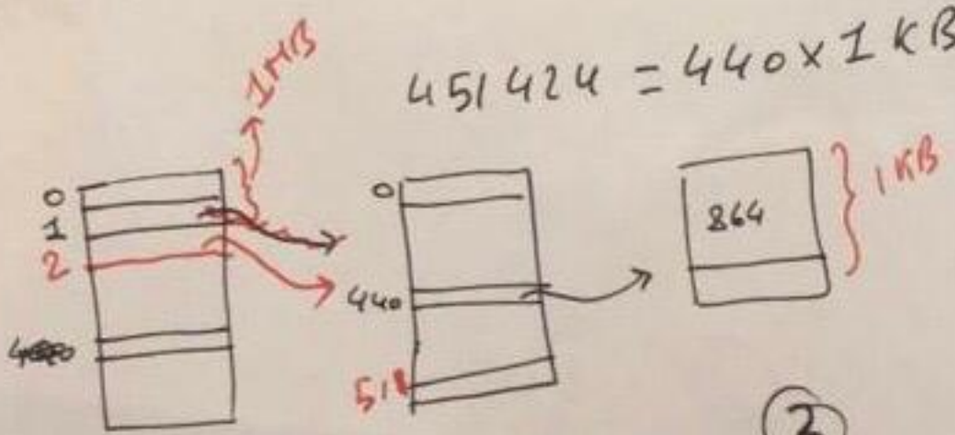
$$2101248 + 4096 + 32 = 2105376 \text{ bytes}$$

$$\text{size(inode)} = \frac{\text{Size(block)}}{\text{Nb of inodes}} = \frac{2^{10}}{32} = \frac{1024}{32} = 32 \text{ bytes} \quad (3)$$

3) file with size = 1500 000 bytes

$$= 1\,MB + 451424$$

$$451424 = 440 \times 1\,KB + 864 \text{ Bytes} \quad (4)$$



(3)

topo [1]

& we need 2 complete map block ≡ 512 to 24 Data block + 440 complete data block + 1 incomplete DB in map 3