UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Departamento de Informática de Sistemas y Computadoras (DISCA)

fSO

EEE1+EEE2 exam
January, 23th 2015

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

SURNAME		**NAME**		Group
IDN		**Signature**		E

- **Keep the exam sheets stapled.**
- **Write your answer inside the reserved space.**
- **Use clear and understandable writing. Answer in a brief and precise way.**
- **The exam has 11 questions, everyone has its score specified.**

---

1. Explain the differences and similarities between a multiprogrammed system and a time sharing system.
(0,5 points)

1	

---

2. An operating system has a Short Term scheduler and a Medium Term scheduler, to each of the following two state transitions give an concrete example when it happens and explain which scheduler deals with it:
(0,5 points)

2	a) A process goes from Execution to Ready
	b) A process goes from Ready to Swap Out (temporary stored in the swapping area).

3. Given the following code that has generated the executable file "Example1"

```
   /*** Example1 ***/
 1 #include "all required headers"
 2
 3 main(){
 4   int i=0,
 5   pid_t pid, pid_x;
 6
 7   for (i=0; i<2; i++) {
 8     pid=fork()
 9     if (pid==0) {
10       sleep(5);
11       pid_x=fork();
12       exit(0);
13     }
14   }
15   sleep(10);
16   while (wait(NULL) =! -1);
17   exit(0);
18 }
```

Answer the following items:

(1,0 point)

3

a) Number of generated processes and their parent ship tree.

b) While the process is running, are **zombie** or **orphan** processes generated?

4. A system has a short term scheduler that uses Round Robin with **q=2 ut**. The system has a single I/O device managed with FCFS. The queue arrival order is: first new processes, next processses coming from I/O and finally processes that leave the CPU. Three processes arrive to the system with the following execution profile and arrival times:

(1.5 = 0,9+0.6 points)

Process	Execution profile	Arrival time
A1	3 CPU + 1 E/S + 1 CPU	0
A2	3 CPU + 1 E/S + 1 CPU	1
B	1 CPU + 3 E/S + 1 CPU	2

a) Fill the following table with the execution time line. (0,9 points)

T	**Ready**	**CPU**	**I/O queue**	**I/O**	**Event**
0					**A1 arrives**
1					**A2 arrives**
2					**B arrives**
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

b) Compute the mean waiting time, mean turnaround time and CPU utilization.

(0,6 points)

4b	Mean waiting time: CPU utilization: Mean turnaround time:

5. Consider the following program with threads and semaphores that has constants N, M and Q defined.

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#define N ...
#define M ...
#define Q ...

pthread_t T[N];
pthread_attr_t atr;
sem_t S1,S2;
int I;

void *P (void *I) {
  sem_wait(&S1);
  printf("Thread is working\n");
  sem_post(&S2);
}

int main() {
  int i;
  pthread_attr_init(&atr);
  sem_init(&S1,0,M); /* Semaphore S1 initialized to M */
  sem_init(&S2,0,0); /* Semaphore S2 initialized to 0 */
  for (i=0;i<N;i++) {
     pthread_create(&T[i],&atr,P,NULL);
  }
  sleep(10);
  for (i=0;i<Q;i++) {
     sem_wait(&S2);
  }
  printf("Bye\n");
  return(0);
}
```

Explain how many threads are created and the messages shown in the standard output when the former code is executed with the following values for constants N, M y Q:

(1,0 point)

5	a) N=10, M=4, Q=2
	b) N=5, M=2, Q=4

6. Threads TREAD1, THREAD2 and THREAD3 are running concurrently, you have to use semaphores with Dijkstra notation (P and V) to guarantee that they will execute functions named with prefix "sequence" in the order suggested by the number at the end of their names. If there are several functions with the same number, no one of them can start before all the functions with the previous number have ended. Furthermore, functions named "sc()" must be executed in mutual exclusion. Indicate what are the initial values of the semaphores used.

(0.75 points)

6

Declare and initialize semaphores

THREAD1 code	THREAD2 code	THREAD3 code

`sequence1();`
 	
`sc();`
 	
`sequence3();`
	`sc();`
	
	`sequence1();`
	
	`sc();`
	
	`sequence3();`
		`sc();`
		
		`sequence2();`
		
		`sequence4();`

7. Explain what can be happening in a multiprogrammed and multiuser system with virtual memory in every one of the following situations:

a) The CPU has a utilization value of 15% and the paging device (hard disk) is busy 97% of time, what those values indicate? how can be improved the CPU utilization?

b) The CPU has a utilization value of 15% and the paging device (hard disk) is busy 15% of time, what those values indicate? how can be improved the CPU utilization?

(0.75 points)

7

a)

b)

8. In a system with Virtual Memory base on demand paging with LOCAL replacement and 64Kbyte page size, two processes A and B are running. Each process has a logical size of 5 pages and their respective page tables (Ref.bit is the reference bit and time values are in decimal) at t=20 have the following content:

Page table for process A				
Frame	Valid bit	Ref. bit	Last access time	Memory loading time
-	i	1	10	1
-	i	1	8	2
0x002	v	0	17	9
0x001	v	1	20	8
0x000	v	1	15	4

Page table for process B				
Frame	Valid bit	Ref. bit	Last access time	Memory loading time
0x004	v	0	18	14
-	i	1	9	6
-	i	1	10	10
0x003	v	1	19	12
-	i	1	11	11

Suppose that at instants t=21 and t=22 the CPU sends logical addresses Process_A:0x00120AB and Process_B:0x00431CB. (1,25 points = 0,25+0,5+0,5)

8

a) Relying on the page tables content, what is the physical address that corresponds to the logical address of process A 0x004247A at t=20?

b) From the data at t=20 explain how will remain the page tables after the addresses sent at t=21 and t=22, applying LRU policy with LOCAL replacement.

Page table for process A				
Frame	Valid bit	Ref.bit	Last access time	Memory loading time

Page table for process B				
Frame	Valid bit	Ref.bit	Last access time	Memory loading time

c) From the data at t=20 explain how will remain the page tables after the addresses sent at t=21 and t=22, applying a Second Chance policy with LOCAL replacement.

Page table for process A				
Frame	Valid bit	Ref.bit	Last access time	Memory loading time

Page table for process B				
Frame	Valid bit	Ref.bit	Last access time	Memory loading time

9. A system has a memory management based on two level paging. It has a 1Mbyte physical memory and every process can address up to 16MBytes. The first level tables contain 16 page descriptors and the page size is 4KBytes. Answer the following questions:

(0.75 points)

9	a) Logical address and physical address formats explaining their fields and number of bits per field.
	b) Compute the internal fragmentation that can be generated when allocating in main memory processes P1, P2 y P3 with sizes: 12524 Bytes, 20480 Bytes and 12289 Bytes, respectively.

10. The following program **ene15.c** counts the number of lines in a file that contain a give word, and writes in the standard output *"The result is XX"* where XX is the number of lines computed. The program works with 3 processes and uses the commands **grep test file** and **wc -l**. The execution syntax is:

```
$ ./ene15 word file
```

```
#include <"all required headers">
int main(int argc, char *argv[]) {
  int fd1[2], fd2[2];
  char result[1000];
  int n;

  pipe(fd1);
  if (fork()==0){ // ********** process 2 *************
    dup2 (fd1[1],STDOUT_FILENO);
    close (fd1[0]); close (fd1[1]);
//TABLE 2
    execlp("grep","grep",argv[1],argv[2],NULL);
  } else {
    close (fd1[1]);
    pipe(fd2);
    if (fork()==0){ // ********** process 3 *************
      dup2 (fd1[0],STDIN_FILENO);
      close (fd1[0]);
      dup2 (fd2[1],STDOUT_FILENO);
      close (fd2[0]); close (fd2[1]);
//TABLE 3
      execlp("wc","wc","-l",NULL);
    } else { // ********** process 1 **************
//TABLE 1
      close(_________);   close(_________);
      n = read(_________,result,1000);
      result[n]=0;
      printf("The result is %s",result);
      close (fd2[0]);
    }
  }
  return 0;
}
```

(1,0 points= 0,3+0,4+0,3)

10

a) Explain the communication squeme between the processes created

b) Obtain the file descriptor tables content at the code points indicated as //TABLE I

	TABLE 1
0	
1	
2	
3	
4	
5	
6	

	TABLE 2
0	
1	
2	
3	
4	
5	
6	

	TABLE 3
0	
1	
2	
3	
4	
5	
6	

c) Complete the variables that are missing in the following code

```
close(          );   close(          );
n = read(       ,result,1000);
```

---

11. A 512Mbyte disk is formatted with a MINIX file system that has the following structure:

Boot block	Super block	i-node bit map	Zone bit map	i-nodes	Data area

The values and sizes used are:

- 64 Byte i-nodes with: 7 direct pointers, 1 indirect and 1 double indirect
- 32 bit (4 Byte) zone pointers
- 16 Byte directory entries
- 1 Block = 1KByte
- 1 zone = 2^2 blocks= 4KByte
- 4096 i-nodes

Obtain:

a) The number of blocks that occupy the i-node bit map, the zone bit map and the i-nodes, respectively.

b) How many different i-nodes have to be consulted to access byte 512 in file */DirX/ArchX*

(1,0 point = 0, 75+0,25)

11

a)

b)