	EXAMEN FINAL SISTEMAS OPERATIVOS	17 de septiembre de 2012
	Nombre y Apellidos: _____	NIF: _____.

Ejercicio 1 (2 pts): Un sistema de ficheros UNIX utiliza bloques de 512 bytes y direcciones de disco de 16 bits. Los **i-nodos** (entradas en una tabla que contiene la información descriptiva de los ficheros) contienen 10 direcciones de disco para bloques de datos, una dirección de bloque índice indirecto simple y una dirección de bloque índice indirecto doble. Conteste de manera razonada a las siguientes cuestiones:

a) (1pt) ¿Cuál es el tamaño máximo de un fichero en este sistema?
b) (1pt) Un programa UNIX crea un fichero en este sistema e inmediatamente después escribe un byte de datos en la posición 1.000 y otro en la posición 10.000. ¿Cuántos bloques de datos ocupa este nuevo fichero en disco?

Bloques 512B, direcciones 2B -> 252 direcciones por bloque

a) Tam. máximo:
   - Por organización: 10+256+(256)^2 -> 10+2^8+2^16 Bloques ~ 32’13MB
   - Por tamaño de índice: 2^16Bloques -> **32MB**

b) Pos 1000 -> 1000/512=1’95 -> está en el bloque directo 1 (empezando en 0)
   Pos 10000 -> 10000/512=19’5 -> está en el bloque directo simple 9

Ejercicio 2 (3 pts): Resolver el siguiente problema de concurrencia:

Supongamos uno o varios hilos productores de Hidrógeno y de Oxígeno y uno o varios hilos creadores de H_2O. Codifique, empleando **mutex** y **variables de condición** como únicos mecanismos de sincronización (avisos y esperas) y exclusión mutua, los procesos productores y consumidor empleando las siguientes variables y funciones para ello:

```
#define MAX_OXIGENO          50
#define MAX_HIDROGENO        100
int nOxigeno=0;
int nHidrógeno=0;
...
```

```
void crearHidrogeno/Oxigeno (){
   while(1){
      //Si puedo (hay hueco)
      crearH/O();
      ...
   }
}
```

```
void crearAgua (){
   while(1){
      //Si hay elementos
      crearH2O();
      ...
   }
}
```

a) (0.5pts) Incluir las variables globales necesarias para el correcto funcionamiento.
b) (1.25pts) Codificar los hilos productores de Hidrógeno y Oxígeno.
c) (1.25pts) Codificar el proceso creador de Agua.

NOTA: No están permitidas las esperas activas y no se debe de hacer ninguna suposición a cerca del número de hilos de cada tipo que hay en el sistema ni del tiempo necesario para la creación de los átomos de H, de O ni del Agua.

```
pthread_mutex_t mutex;
pthread_cond_t vCond;
```

```
void crearHidrogeno(){
  while(1){
    crearH();
    lock(mutex);
    while(nHidrógeno==MAX_HIDROGENO){
      cond_wait(vCond, mutex)
    nHidrógeno++;
    //Alguien puede estar bloqueado
    if(nHidrógeno==1)
      cond_signal(vCond);
    unlock(mutex);
    }
  }
}
// El de Oxígeno igual:
  nHidrógeno     -> nOxigeno
  MAX_HIDROGENO -> MAX_OXIGENO
```

```
void crearAgua (){
  while(1){
    lock(mutex);
    while(nHidrógeno<2||nOxigeno<1)
      cond_wait(vCond, mutex);
    nHidrógeno-=2;    nOxigeno--;
    //Alguien puede estar bloqueado
    if(nHidrógeno==MAX_HIDROGENO-2 ||
       nOxigeno == MAX_OXIGENO-1)
       cond_broadcast(vCond);
    unlock(mutex);
    crearH2O();
    }
  }
}
```

Ejercicio 3 (2 pts): Supongamos un sistema cuyo **TLB** (Translation Look-aside Buffer) y **TP** (Tabla de Páginas) contiene la siguiente información:

TLB

Página	Marco
4	2
15	3
12	4
7	9

TP

Página	Marco	Página	Marco
0	Null	8	1
1	15	9	6
2	7	10	Null
3	Null	11	Null
4	2	12	4
5	Null	13	Null
6	Null	14	Null
7	9	15	3

Suponga que el proceso activo genera la siguiente traza de direcciones virtuales:

0x4A, 0x40, 0xC7, 0x10, 0x43, 0xAF

a) (1.5pts) Indicar cuáles generan acierto o fallo en el TLB y en la TP así como los cambios que hace el SO en ambas tablas. ¿Cuáles son las direcciones físicas tras la conversión?

b) (0.5pts) Suponga que un acceso al TLB supone 2ns, un acceso a la TP 50ns y un fallo de página 5ms. ¿Cuánto tiempo tardan en servirse las peticiones 0x40, 0x10 y 0xAF empleando los valores iniciales de las tablas?

NOTA: Las entradas del TLB están ordenadas de más antiguas a más nuevas, siendo la más antigua la primera (Pag. 4 -- Marco 2), en caso de reemplazamiento usar el algoritmo FIFO.

- 0x4A -> acierto en el TLB, dir física 0x2A
- 0x40 -> acierto en el TLB, dir física 0x20
  - Tiempo 2ns
- 0xC7 -> acierto en el TLB, dir física 0x47
- 0x10 -> fallo TLB, acierto TP, dir física 0xF0
  - saco [4-3] y meto [1-15] al final del TLB
  - tiempo 2ns + 50ns
- 0x43 -> fallo TLB, acierto TP, dir física 0x23
  - saco [15-3] y meto [4-2] al final del TLB
- 0xAF -> fallo TLB y TP, nuevo marco->**(X)** dir física 0x**(X)**F
  - saco [12-4] y meto [10-**(X)**]
  - tiempo 2ns + 50ns + 5ms

Ejercicio 4 (3 pts): Responda a las siguientes preguntas sobre el código que se presenta a continuación:

a) (2pts) Dibuje el estado de las tablas de descriptores de ficheros abiertos y la tabla intermedia del SO justo antes de la finalización del proceso hijo. Asuma que el proceso padre es más prioritario que el hijo.

b) (1pt) Indique el contenido final de los ficheros `examen` y `result`.

```
int fd1;
int fd2;
char buf1[2000]="aaaa...a";
char buf2[1500]="bbbb...b";
```

```
main() {
  fd1=open("/tmp/examen",O_CREATE...);
  if (fork() == 0) {
      hijo();
      exit(0);
  }
  else {
    //Escribimos 1500 c's en buf2
    memcpy(buf2,"cccc...c",1500);
    fd2=open("/tmp/result",O_RDWR);
    wait(NULL);
    lseek(fd1,1500,SEEK_CUR);
    write(fd1,buf2,500);
    write(fd2,buf2,500);
  }
}
```

```
void hijo( ) {
  fd2=open("/tmp/result",O_RDWR);
  write(fd1,buf1,2000);
  lseek(fd1,2500,SEEK_CUR);
  write(fd1,buf1,500);
  write(fd2,buf2,1500);
  close(fd2);
}
```

Tabla FD Padre	Tabla FD Hijo
STDIN	STDIN
STDOUT	STDOUT
STDERR	STDERR
fd1 -> id1	fd1 -> id1
fd2 -> id2	fd2 -> id3

Tabla SO		
Id	Puntero LE	Fichero
id1	5000	examen
id2	0	result
id3	1500	result

Contenido de `examen`

Posición	0-1999	2000-4499	4500-4999	5000-6499	6500-7999
Contenido	aaa….aaa	vacío	aaa…aaa	vacío	ccc…ccc

Contenido de `result`

Posición	0-499	500-1499
Contenido	ccc…ccc	bbb….bbb