	EXAMEN FINAL SISTEMAS OPERATIVOS	**29 de enero de 2013**
	Nombre y Apellidos: _____	NIF: _____.

Ejercicio 1 (3 pts): Una empresa de reparto recibe distintos tipos de paquetes que debe distribuir a través de repartidores especializados, cada repartidor es capaz de repartir sólo un único tipo de paquete.

1. (1pt) Implemente un hilo ***Recepcionista()*** encargado de recibir los paquetes y avisar a los hilos ***Repartidor(int tipo)*** según el siguiente esquema y utilizando como mecanismo de sincronización únicamente semáforos:

```
#define TIPO1           1
#define TIPO2           2
#define MAX_PAQUETES    100
int nTipo1=0, nTipo2=0;
//Resto de variables necesarias (valor inicial)

Recepcionista () {
   int tipo;
   while(1){
      //Si el almacén está lleno, espera a que haya hueco
      tipo=atenderCliente(); //Tiempo de espera largo
      switch(tipo){
      case TIPO1:
         nTipo1++;
         //Avisar a repartidores tipo 1
         break;
      case TIPO2:
         nTipo2++;
         //Avisar a repartidores tipo 2
         break;
      }
   }
}
```

2. (1pt) Implemente un hilo genérico ***Repartidor(int tipo)*** capaz de funcionar adecuadamente con el ***Recepcionista***.
3. (1pt) ¿Qué tipo de semáforos **POSIX** podríamos usar si en vez de hilos usásemos procesos emparentados? ¿y no emparentados? Razone sus respuestas.

Ejercicio 2 (2 pts): Un disco tiene 100 cilindros, 10 cabezas. Cada pista está compuesta por 20 sectores de 512 bytes cada uno. El disco gira a 3000rpm y tiene un mecanismo de cabezas móviles con un tiempo de búsqueda de 0.5ms. Dada la siguiente cola de peticiones de sectores pendientes en un disco, en formato CHS (cilindro, cabeza y sector):

A(90,5,5); B(10,9,6); C(90,5,6); D(90,7,6); E(10,9,5); F(90,7,7)

y teniendo en cuenta que las cabezas están posicionadas inicialmente en el cilindro 20 y anteriormente en el 19, responda a las siguientes preguntas:

1. (0.5pts) Calcule la capacidad del disco.
2. (1pt) Determinar el orden de acceso al aplicar las políticas **FIFO**, **SSTF**, **SCAN** y **C-SCAN**.
3. (0.5pts) Calcular el tiempo en ir desde **A** hasta **B** directamente.

Ejercicio 3 (2 pts): Un sistema de paginación pura tiene un tamaño de página de 32 bytes, una memoria virtual de 8 páginas y 5 marcos asignados (del 0 al 4). El contenido actual de la tabla de páginas es el siguiente:

	Marco	Validez
0	4	1
1	4	0
2	1	1
3	3	1
4	2	1
5	3	0
6	0	1
7	3	0

1. (1pt) En caso de ser válidas, ¿qué direcciones físicas referencian las siguientes direcciones virtuales? y en caso de ser inválidas, ¿qué ocurriría?: `0x1F, 0x20, 0x93, 0xFF`
2. (1pt) En un momento dado, un proceso tiene `PC=0xD0`, ¿cuántos accesos a memoria, como mínimo, se producen si en dicha dirección se encuentra la instrucción: `load r0, 0x20`?

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

1. (1pts) Dibuje el estado de las tablas de descriptores de ficheros abiertos y la tabla intermedia del SO justo antes de que el hijo haga `close(fd2)`.
2. (1pt) Indique el contenido de los ficheros `test1` y `test2`.
3. (2 puntos). El sistema de ficheros usado es una simplificación del modelo de Unix: nodos-i con 2 punteros directos, un indirecto simple y uno doble. Cada puntero ocupa 4 bytes, y los bloques son de 512B. Describa el contenido del nodo-i y de los bloques de datos del fichero `test1` tras la ejecución de este código. Asuma que, en caso de necesitar bloques de datos, se asignan partir del 500 y en orden ascendente.

```
int fd1, fd2;
char cad1[2048]="1111...1";
char cad2[512] ="2222...2";
char cad3[5120]="3333...3";
```

```
main() {
  fd1=open("/test1",O_CREATE...);
  if (fork() == 0) {
     hijo();
     exit(0);
  }
  else {
    fd2=open("/test1",O_RDWR);
    wait(NULL);
    read (fd2,cad3,1024);
    write(fd2,cad3,2048);
  }
}
```

```
void hijo( ) {
  write(fd1,cad1,1024);
  fd2=open("/test2",O_RDWR);
  lseek(fd1,5*1024,SEEK_CUR);
  write(fd1,cad2,512);
  write(fd2,cad3,4*1024);
  close(fd2);
}
```