

Apellidos: \_\_\_\_\_ Nombre \_\_\_\_\_

### **Cuestiones (3 puntos)**

#### **Cuestión 1: (0,5 pt)**

Indicar qué imprime el siguiente programa si se ejecuta desde el intérprete de órdenes y considerando que el identificador de proceso del intérprete es 1424, el de este proceso es el 1430 y el del hijo el 1445. Se considera también que no se produce ningún error en el fork.

```
1445 1430
1430 1424
```

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

void main(void)
{
    pid_t pid;

    pid = fork();
    switch(pid)
    {
        case -1:
            perror("fork");
            break;
        case 0:
            printf(" %d; %d \n", getpid(), getppid());
            break;
        default: /* padre */
            printf(" %d; %d \n", getpid(), getppid());
    }
}
```

#### **Cuestión 2: (1 pt)**

Rellenar el siguiente programa para que el proceso ignore la señal SIGINT que se genera cuando se pulsa CTRL-C y a continuación ejecute un bucle infinito.

```
#include <stdio.h>
#include <signal.h>

void main(void)
{
    struct sigaction act;

    act.sa_handler= SIG_IGN; /* Ignora la señal */
    act.sa_flags= 0; /* Ninguna acción especial */
    sigemptyset(&act.sa_mask); /* Ninguna señal bloqueada */

    sigaction(SIGINT, &act, NULL);

    while(1); /* Bucle infinito */
}
```

¿Cómo se puede terminar?

```
^z para parar el proceso
kill -SIGKILL pid para terminarlo
```

#### **Cuestión 3: (0,5 pt)**

En un sistema de memoria virtual con paginación, con una memoria real de 256 Mbytes ( $2^{28}$ ), 32 bits para la dirección virtual y páginas de 4Kbytes ( $2^{12}$ ), indicar:

Número de marcos de página:  $2^{28}/2^{12} = 2^{16}$  **marcos de página (64K marcos)**  
Número de páginas virtuales:  $2^{32} / 2^{12} = 2^{20}$  **páginas (1 M páginas)**

Apellidos: \_\_\_\_\_ Nombre \_\_\_\_\_

#### Cuestión 4: (1 pt)

Para el vector de recursos existentes  $E=(4,6,7,3,2)$ , para la siguiente matriz de recursos asignados (MRA) y la siguiente matriz de recursos que se necesitan todavía (MRN):

MRA:

	R1	R2	R3	R4	R5
P1	1	0	3	0	1
P2	1	2	1	0	0/1
P3	1	1	2	1	0

MRN:

	R1	R2	R3	R4	R5
P1	3	1	1	2	0
P2	1	0	1	0	1/0
P3	2	4	1	2	0

Si el proceso P2 solicita el recurso R5, ¿es adecuado o no concederle el recurso? Justificar la respuesta.

SI

NO

Si al conceder R5 a P2 el estado es seguro, es adecuado, en caso contrario, no sería adecuado. Para mirar si el estado es seguro se aplica el algoritmo del banquero para múltiples recursos:

$E = (4\ 6\ 7\ 3\ 2)$       P se obtiene de la suma por columnas de MRA  
 $P = (3\ 3\ 6\ 1\ 1)$        $E - P = D$   
 $D = (1\ 3\ 1\ 2\ 1)$

Si concedemos el recurso R5 a P2, vemos el estado al que llegamos (señalado en las matrices):

$P = (3\ 3\ 6\ 1\ 2)$   
 $D = (1\ 3\ 1\ 2\ 0)$

Aplicamos el algoritmo:

$MRN(P2) < D \rightarrow$  termina P2  $D = D + (1\ 2\ 1\ 0\ 1) = (2\ 5\ 2\ 2\ 1)$   
 $MRN(P3) < D \rightarrow$  termina P3  $D = D + (1\ 1\ 2\ 1\ 0) = (3\ 6\ 4\ 3\ 1)$   
 $MRN(P1) < D \rightarrow$  termina P1  $D = D + (1\ 0\ 3\ 0\ 1) = (4\ 6\ 7\ 3\ 2)$

Se ha podido llegar a terminar todos los procesos. Concluimos que el estado es seguro. Por lo tanto es adecuado conceder el recurso R5 a P2.