

Començat el	-
Estat	Acabat
Completat el	-
Temps emprat	-
Punts	5,00/5,00
Qualificació	10,00 sobre 10,00 (100%)

Pregunta 1
Correcte

Puntuat 1,00
sobre 1,00

Dado el siguiente código, suponiendo que ninguna llamada a sistema provoca error, ¿qué podemos asegurar con respecto al SIGALRM que el sistema operativo enviará como consecuencia del alarm(2) que aparece en el código?

```
void funcion(int signum) {
char buf[80];
sprintf(buf, "SIGALRM!\n");
write(1,buf, strlen(buf));
}

main(int argc, char *argv[]){
    struct sigaction trat;

    ...
    trat.sa_handler = funcion;
    sigaction(SIGALRM, &trat, NULL);
    alarm(2);
    fork();
    execlp("./A","A",(char *) NULL);
    waitpid(-1,NULL,0);
}
```

Trieu-ne una o més:

- ☐ a. Si el proceso hijo recibe un SIGALRM mientras está ejecutando el programa "A" siempre escribirá el mensaje "SIGALRM!\n"
- ☐ b. Si el proceso hijo recibe el SIGALRM (debido al alarm(2)) mientras está ejecutando el código del programa "A", si ese código no ha reprogramado el tratamiento del SIGALRM, el proceso hijo morirá
- ☐ c. El proceso padre siempre escribirá el mensaje "SIGALRM!\n" al cabo de 2 segundos de la ejecución del alarm(2)
- ☐ d. Si el proceso padre recibe el SIGALRM mientras está ejecutando el programa "A" siempre escribirá el mensaje "SIGALRM!\n"
- ☒ e. Si el proceso padre recibe el SIGALRM mientras está ejecutando el código del programa "A", si ese código no ha reprogramado el tratamiento del SIGALRM, el proceso padre morirá
- ☐ f. Podría ocurrir que el proceso hijo recibiera el SIGALRM, debido al alarm(2), antes de ejecutar el execlp y entonces ejecutaría el tratamiento por defecto y moriría
- ☐ g. Lo más probable es que el proceso padre reciba el SIGALRM mientras está ejecutando el waitpid y entonces escribirá el mensaje "SIGALRM!\n"
- ☒ h. El alarm(2) que tenemos en este código no provoca que el hijo reciba un SIGALRM

Les respostes correctes són: Si el proceso padre recibe el SIGALRM mientras está ejecutando el código del programa "A", si ese código no ha reprogramado el tratamiento del SIGALRM, el proceso padre morirá, El alarm(2) que tenemos en este código no provoca que el hijo reciba un SIGALRM

Pregunta**2**

Correcte

Puntuat 1,00
sobre 1,00

Un proceso crea 10 procesos hijos. Por su parte, el padre controla la finalización de los mismos con el tratamiento del signal SIGCHLD.

En qué escenario el proceso padre puede recibir las señales SIGCHLD procedentes de sus hijos sin que se solape el tratamiento de las mismas?

Trieu-ne una o més:

- ☐ a. Tratando la señal SIGCHLD entre la creación de un hijo y el siguiente
- ☒ b. Poniendo un sigsuspend (con la máscara vacía) en el padre entre la creación de un hijo y el siguiente
- ☒ c. Cuando los hijos se les obliga a ejecutar secuencialmente

Les respostes correctes són: Cuando los hijos se les obliga a ejecutar secuencialmente, Poniendo un sigsuspend (con la máscara vacía) en el padre entre la creación de un hijo y el siguiente

Pregunta**3**

Correcte

Puntuat 1,00
sobre 1,00

Asumiendo que antes del bucle hemos reprogramado el tratamiento del signal SIGINT para que ejecute la función func, ¿cuántos mensajes aparecerán en pantalla?

```
void func(int s){
    printf("Signal %d\n", s);
}

int main(int argc, char **argv){
    ...
    for(i=0;i<3;i++){
        if (fork()==0){
            kill(getppid(), SIGINT);
            exit(0);
        }
    }
}
```

Trieu-ne una o més:

- ☐ a. Siempre 3 mensajes porque el proceso recibe 3 veces el SIGINT
- ☒ b. Puede suceder que el proceso padre reciba ninguna vez el signal
- ☒ c. No lo podemos saber con certeza, porque se pueden solapar varios SIGINT antes de su tratamiento
- ☐ d. Podemos asegurar que como mínimo se mostrará una vez el mensaje

Les respostes correctes són: No lo podemos saber con certeza, porque se pueden solapar varios SIGINT antes de su tratamiento, Puede suceder que el proceso padre reciba ninguna vez el signal

Pregunta**4**

Correcte

Puntuat 1,00
sobre 1,00

Quan un procés executa un exec(), què succeeix amb els signals pendents?

Trieu-ne una o més:

- ☒ a. Els signals continuen pendents i quan es tractin les reprogramades no tindran el mateix comportament que abans de l'exec()
- ☐ b. Els signals continuen pendents i quan es tractin les reprogramades tindran el mateix comportament que abans de l'exec()
- ☐ c. Els signals pendents es perden quan es fa exec()
- ☒ d. Si estaven bloquejats abans de mutar, continuant bloquejats després de mutar

Les respostes correctes són: Els signals continuen pendents i quan es tractin les reprogramades no tindran el mateix comportament que abans de l'exec(), Si estaven bloquejats abans de mutar, continuant bloquejats després de mutar

Pregunta **5**

Correcte

Puntuat 1,00
sobre 1,00

Dos procesos que no están emparentados (padre-hijo), ¿Pueden enviarse signals?

Trieu-ne una:

- ☐ a. No
- ☒ b. Si

La resposta correcta és: Si

◀ test sesión 3

Salta a...

session 4 test ▶