Uso de “Real time Signals”

C Prácticas

Las usaremos para asegurar que todas las señales de los hijos avisando de que ya han apostado llegan al padre.

El problema que hay con las señales normales es que si varios procesos envían simultáneamente (casi simultáneamente, vaya) señales del mismo tipo a un proceso (en nuestro caso el padre), cuando el S.O. decide pasar a ejecución al padre, si tiene varias señales del mismo tipo pendientes para ese proceso, las une en una sola y lanza al manejador del padre para ese tipo de señales como si solo hubiese tenido una disponible.

Por ese motivo, se pueden perder algunas de las señales. Nosotros utilizamos el manejador del padre para contar el número de señales recibidas de los hijos antes de seguir (generando la combinación ganadora). Si se da el caso que el S.O. une varias señales en una, nuestro contador no llegará al número de hijos creados, con lo que no saldremos del bucle[1] y el programa se cuelga.

Las señales de tiempo-real no se pierden, se llama siempre al manejador por cada una de ellas, el sistema operativo las encola siempre, y si no puede encolarlas da un error[2], cosa que no hace con las señales normales.

Para enviar una señal normal utilizaremos kill(...) para una señal en tiempo real sigqueue(...).

De igual forma, para definir los handlers de las señales utilizaremos sigaction(..) en vez de signal(...).

Definiremos una estructura para establecer la acción (manejador) y sus propiedades para la señal:

```
struct sigaction sa;
```

Cuando definamos el manejador de la señal que recibe el padre (y que indica que el hijo ya ha apostado) haremos

```
ssa.sa_sigaction = handler;
sa.sa_flags = SA_SIGINFO;
sigaction(SIGRTMIN,&sa,NULL);
```

... donde handler es el nombre de nuestra función manejadora. Debemos definir el flag SA_SIGINFO si queremos enviar información adicional junto con nuestra señal en tiempo real.[3]

Si os fijáis la señal que recibirá el padre será la señal SIGRTMIN que es una macro con el valor de la señal de tiempo real con valor mínimo. Existe también la SIGRTMAX con el valor máximo. Una señal es de tiempo real si su valor está entre SIGRTMIN y SIGRTMAX inclusive. Definidos en signal.h

Por tanto nuestro envío desde el código del hijo con sigqueue(...) será con la señal SIGRTMIN.

Definiremos una unión de tipo sigval para almacenar la información adicional que el hijo le envía al padre junto con la señal SIGRTMIN

```
union sigval v;
```

... siendo esta la unión[4] :

```
union sigval {
 int sival_int;
 void *sival_ptr;
 }
```

Cuando desde el código del hijo queramos enviar la señal haremos lo siguiente:

```
v.sival_int = getpid(); //asignamos valor al dato
 sigqueue(getppid(),SIGRTMIN,v); //enviamos la señal de tiempo real
```

Os coloco el código de un manejador de ejemplo para la señal SIGRTMIN con datos adjuntos[5] :

```
void handler (int signo, siginfo_t *info, void *other) {
 char *src;
 switch(info->si_code) {
 case SI_QUEUE: src = "sigqueue"; break;
 case SI_TIMER: src = "timer"; break;
 case SI_ASYNCIO: src = "asyncio"; break;
 case SI_MESGQ: src = "msg queue"; break;
 case SI_USER: src = "kill/abort/raise"; break;
 default: src = "unknown!"; break;
 }
 printf("Signal %d: source = %s, data = %d\n",signo, src, info->si_value.sival_int);
 }
```

Si compiláis y todo va bien.... ya no se pierden señales de los hijos, con lo que solucionamos el problema.

---

1. El bucle que comprueba si señales_recibidas<numero_hijos y que tiene un pause() como único código.
2. Caso de que el buffer o cola de señales esté lleno, no se dará ese caso para nuestra práctica pues el número de hijos lo limitamos a 10
3. Esto realmente no lo vamos a necesitar, pues nosotros contaremos simplemente las señales, pero se puede utilizar para enviar el pid del hijo que envía la señal, por ejemplo. Podéis ponerlo y enviar el id como en los ejemplos que os envío o poner 0 como dato...
4. No tenéis que definirla, ya lo está en <signal.h> o en <sys/types.h>
5. En negrita pongo lo que realmente necesitáis, lo otro es para ver tipo de origen de la señal....

← Entrada anterior

Entrada siguiente →

Escuela Politécnica Superior de Elche

Universidad Miguel Hernández

Miguel Onofre Martínez Rach