Auxiliar ? - CC3301

June 4, 2010

Problema 1 (Señales)

- (i) Escriba un programa en C llamado `cronometro.c`, que al ser llamado espere hasta que el usuario tipee un `control-C` (señal `SIGINT`). Al recibir la señal, el programa debe imprimir el tiempo transcurrido (en segundos) desde que comenzó su ejecución, y luego terminar. Nota: Recuerde que la función `time(NULL)` default entrega el número de segundos transcurridos desde el Epoch.
- (ii) Queremos contar el tiempo (en segundos) que demora una función en retornar. La idea es que, para contar cuantos segundos la función `fun()` en ejecutar hacemos:

```
int secs;

start_timer();
fun();
/* retorna los segundos pasados desde start_timer()*/
secs = stop_timer();
```

Usando señales e interrupciones de reloj, implemente `start_timer()` y `stop_timer()`. Explique que pasa si dentro de `fun()` alguien decide usar esas funciones para medir el tiempo de otra función y proponga una solución para que funcione bien.

Problema 2 (fork/pipes)

Queremos comparar el output de dos funciones `char foo(char* w)`, `char bar(char *w)` en paralelo. Las funciones `foo` y `bar` no retornan inmediatamente todo el output, si no que van retornando caracter a caracter. Por ejemplo

```
/* Supongamos que foo("hola") es "chao"*/
char w1 = foo("hola"); /* w1 es 'c' */
char w2 = foo(NULL); /* w2 es 'h' */
char w3 = foo(NULL); /* w2 es 'a' */
```

```
char w4 = foo(NULL); /* w2 es 'o' */  
char w5 = foo(NULL); /* w2 es '\0' */
```

Para ello un proceso padre deberá crear dos procesos hijos que le envían a través de un pipe los valores que va entregando la función