Sistemas Operativos

Proyecto II

Javier García Algarra
javier.algarra@u-tad.com
Miguel Ángel Mesas
miguel.mesas@u-tad.com
Carlos M. Vallez
carlos.vallez@u-tad.com

Descripción

En este proyecto vamos a desarrollar una aplicación para encontrar números primos paralelizando la tarea entre varias instancias de un mismo proceso.

Recordamos que un número primo es un número natural mayor que 1 que solo es divisible por sí mismo y por la unidad. Hay numerosos algoritmos para comprobar si un número es primo, pero para este usaremos uno de fuerza bruta, para poder comprobar el efecto de la paralelización.

Un número K es primo si resulta divisible por cualquier cifra entre 2 y K/2. El algoritmo simplemente prueba a dividir entre 2 y K/2 hasta encontrar un número por el que es divisible (no es primo) o finalizar el bucle sin conseguirlo (es primo).

Descripción

Por claridad vamos a identificar tres papeles del mismo proceso como RAIZ, SERVER y CALCulador, pero debe tenerse en cuenta que todas están en el mismo ejecutable. Se desarrollará un único fuente llamado `encuentraprimos.c`

El proceso inicial (RAIZ) se clona con `fork()` y al hijo lo llamamos proceso SERVER. Este, a su vez creará tantos hijos como se indique por línea de comandos. Cada uno de ellos es un proceso CALCulador.

El SERVER crea una cola de mensajería SYSTEM V, que es única y sirve para la comunicación bidireccional del SERVER con los CALCuladores. Esa cola no interviene en la comunicación RAIZ SERVER.

SERVER pide memoria dinámica para guardar una lista con las identidades de los CALCuladores.

Descripción

Cada CALCulador envía un mensaje al server indicando que ya está listo y cual es su pid. El proceso SERVER muestra por pantalla la jerarquía de procesos:

```
RAIZ          SERV          CALC
17001         17002         17003
                            17004
                            17005
                            17006
```

Cuando ha recibido todos los mensajes de inicio (COD_ESTOY_AQUI), el SERVER devuelve con otro mensaje a cada proceso CALCulador, el inicio del rango en el que debe empezar a buscar primos y hasta donde debe llegar.

Descripción

Los procesos hijos buscan los números primos con el algoritmo de fuerza bruta. Cada vez que localizan uno, envían un mensaje al SERVER con su identidad y el número hallado. El SERVER presenta un texto por el terminal indicando qué hijo ha localizado un primo y cual es el número. Esta salida se puede silenciar con el valor de verbosity proporcionado por comando. Adicionalmente se escribe el valor del número primo en el fichero `primos.txt`

Si el número total de primos en un momento dado es múltiplo de 5 escribe esa cifra en `cuentaprimos.txt`. Este fichero servirá para el funcionamiento del proceso RAIZ Cuando uno de los CALCuladores termina de explorar el rango indica al SERVER que ha finalizado la tarea con el código correspondiente en un mensaje.

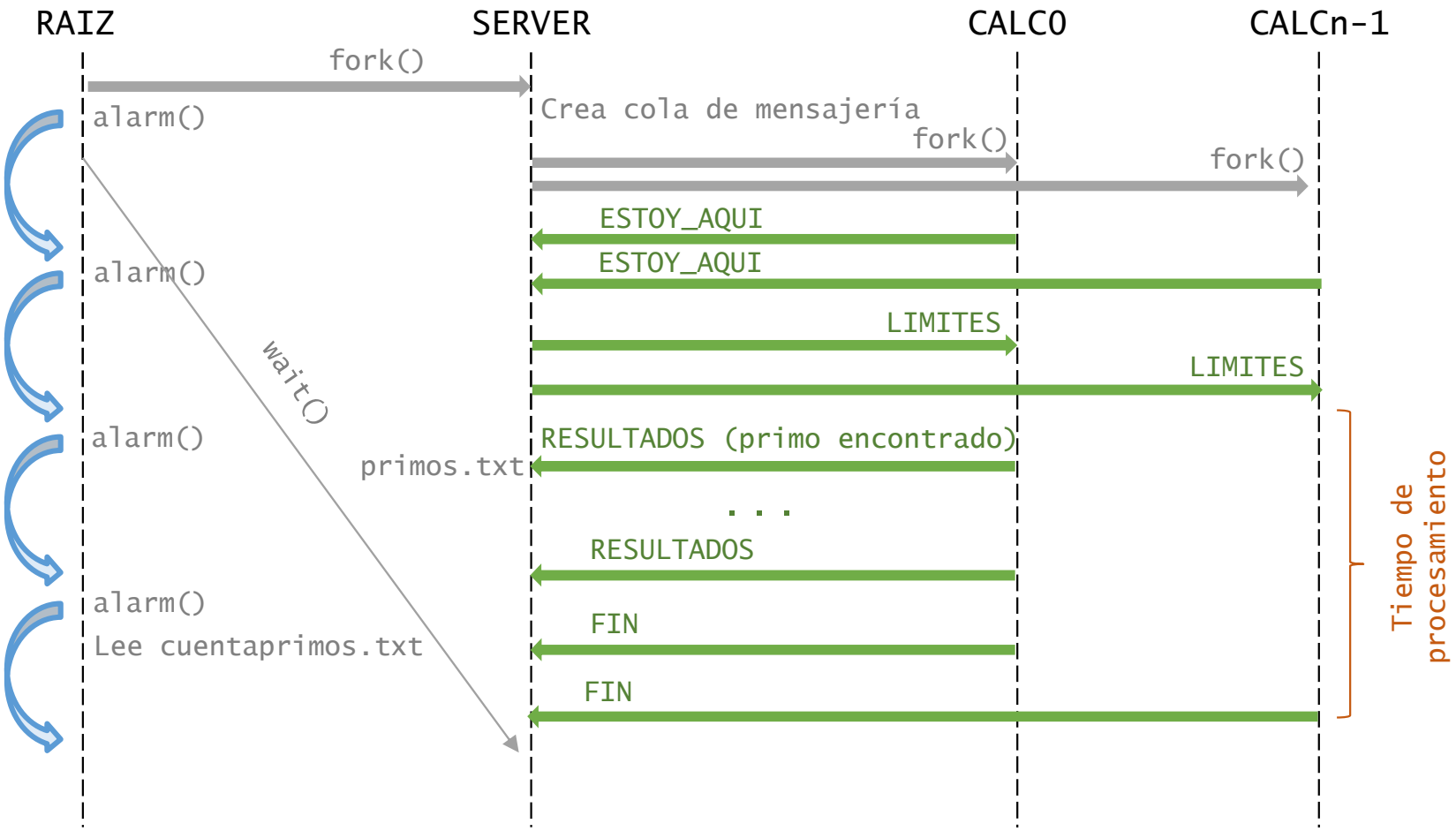
Descripción

Al recibir el mensaje de finalización de todos los hijos, el SERVER cierra el fichero `primos.txt` y muestra por pantalla el tiempo transcurrido. El comportamiento del proceso raíz es el siguiente. Tiene que arrancar un `alarm()` con periodo de 5 segundos y armar un manejador de interrupción. Cada vez que vence, lee el fichero `cuentaprimos.txt` y muestra su contenido por pantalla, con independencia del valor de verbosity.

Cuando termine el proceso SERVER, lo que detectará con el `wait()` oportuno, cuenta el número de líneas del fichero `primos.txt` para indicar cuántos primos se han localizado. Un script llamado `proyectoSSOO.sh` que se proporciona con la práctica se encargará de arrancar el proceso `cuentaprimos` y ordenar el fichero de resultados.

Descripción

¡Ojo! Todos son instancias de encuentraprimos



Evaluación

- El proceso controla correctamente los parámetros de entrada (0,5 puntos)
- Los CALC envían el mensaje de ESTOY_AQUI y el SERVER presenta la jerarquía de procesos (2 puntos)
- Los procesos CALC envían cada número primo localizado con un mensaje, el SERVER lo interpreta correctamente y actualiza los ficheros primos.txt y cuentaprimos.txt (2 puntos)
- El SERVER muestra los números por pantalla o los oculta en función de verbosity (0,5 puntos)
- RAIZ maneja correctamente la temporización con alarm() y muestra el contenido de cuentaprimos.txt cada 5 segundos (2 puntos).
- El contenido de sorterprimos.txt es correcto (1 punto).
- Limpieza del código, legilibilidad, documentación y commits bien comentados (2 puntos)
- SOLO SE ENTREGA UNA PRÁCTICA POR EQUIPO y el código debe estar disponible para el profesor en github.
- TIENES QUE USAR LOS #defines, #includes y prototipos de función que figuran en esqueleto.c Puedes añadir, pero no quitar ni modificar.