

Práctica 1

Sistemas Operativos

Víctor García Carrera, Alfonso Carvajal Moreno de Barreda

En la entrega de esta práctica se incluye un makefile para generar los ejecutables de los ejercicios a entregar, la implementación en C de los mismos, la documentación generada con la herramienta Doxygen, un fichero de texto llamado datospareja con nuestra información básica y este pdf con las explicaciones pertinentes de los entregables.

Ejercicio4:

Apartado a)

Aquí el resultado de los árboles obtenidos al modificar el ejercicio anterior. Existen procesos huérfanos.

```
PADRE 7154
HIJO 7154 con padre 7153
PADRE 7156
HIJO 7156 con padre 7154
PADRE 7158
HIJO 7158 con padre 7156
e4a(7154)└─e4a(7156)└─e4a(7158)
          │   └─sh(7159)──pstree(7162)
          └─sh(7157)──pstree(7160)
PADRE 7163
HIJO 7163 con padre 7153
PADRE 7165
HIJO 7165 con padre 7163
e4a(7156)└─e4a(7158)
          └─sh(7159)──pstree(7162)
PADRE 7169
HIJO 7169 con padre 7154
e4a(7158)
e4a(7163)└─e4a(7165)
          └─sh(7167)──pstree(7168)
PADRE 7172
HIJO 7172 con padre 7153
e4a(7165)
e4a(7169)
e4a(7172)
```

Apartado b)

Modificamos el ejercicio anterior. No existen procesos huérfanos al haber utilizado la función wait() en la implementación.

```
PADRE 7310
HIJO 7310 con padre 7309
PADRE 7312
HIJO 7312 con padre 7310
PADRE 7316
HIJO 7316 con padre 7312
e4b(7312)———e4b(7316)
      |
      └─sh(7317)——pstree(7318)
PADRE 7319
HIJO 7319 con padre 7310
e4b(7316)
e4b(7310)———e4b(7312)———e4b(7316)
      |
      └─sh(7317)
PADRE 7322
HIJO 7322 con padre 7309
PADRE 7324
HIJO 7324 con padre 7322
e4b(7319)
e4b(7322)———e4b(7324)
      |
      └─sh(7325)——pstree(7327)
PADRE 7328
HIJO 7328 con padre 7309
e4b(7324)
e4b(7328)
```

Ejercicio5:

Apartado a)

Todos tienen solo 1 hijo y no acaban hasta que acaba su hijo. En este el padre espera al hijo y cuando el hijo acaba, el padre también. El wait() está dentro del bucle y el proceso hijo termina solo cuando se sale del bucle, es decir, cuando todos han sido creados.

```
PADRE 6845
HIJO 6846 con padre 6845
PADRE 6846
HIJO 6847 con padre 6846
PADRE 6847
HIJO 6848 con padre 6847
```

This process IDS == 6848 is ending

This process ID == 6847 is ending

This process ID == 6846 is ending

This process ID == 6845 is ending

Apartado b)

Todos tienen el mismo padre. En este el padre espera al hijo y cuando el hijo acaba lanza otro proceso. El wait() está dentro del bucle, y el hijo termina nada más ejecutarse dentro de la condición.

HIJO 6854 con padre 6853

This process ID == 6854 is ending

PADRE 6853

HIJO 6855 con padre 6853

This process ID == 6855 is ending

PADRE 6853

HIJO 6856 con padre 6853

This process ID == 6856 is ending

PADRE 6853

This process IDS == 6853 is ending

Ejercicio6:

Hay que liberar memoria en ambos. El padre no puede acceder a la información almacenada por el hijo.

Introduce un nombre: hello

Este es el caso en el que intentamos hacerlo y da errores de valgrind.

HIJO === 5798

BUF == hello

PADRE == 5797

BUF ==

Ejercicio 8:

Todo lo necesario para comentar se encuentra en su implementación.

Ejercicio 9:

Todo lo necesario para comentar se encuentra en su implementación.