SISTEMAS OPERATIVOS: AYUDANTÍA 3

Ayudante: Yerko Ortiz

Objetivo de la ayudantía: Reforzar conceptos y contenidos referentes a procesos(fork y pipes).

Compilador mental

Compile y ejecute los siguientes algoritmos a mano, describa las salidas correspondientes y dibuje el árbol de procesos asociado.

Algoritmo 1

```c
#define WAIT_TREE for(; wait(NULL) > 0;)
int algoritmo1(void)
{
        printf("L0\n");
        fork();
        printf("L1\n");
        fork();
        printf("Bye\n");
        WAIT_TREE;
}
```

- **Output:**
- **Árbol:**

Algoritmo 2

```c
#define WAIT_TREE for(; wait(NULL) > 0;)
void algoritmo2(void)
{
    if (fork() || fork())
        fork();
    printf("1 ");
    WAIT_TREE;
}
```

- **Output:**
- **Árbol:**

Algoritmo 3

```c
#define WAIT_TREE for(; wait(NULL) > 0;)
#define MAX_HEIGHT 2
void processTree(int current_height)
{
    if (current_height > MAX_HEIGHT)
        return;
    if (!fork() || !fork())
       processTree(current_height + 1);
    else {
        printf("OWN PID: %d | PARENT PID: %d\n",
            getpid(), getppid());
        WAIT_TREE;
    }
}
```

- **Output:**
- **Árbol:**

> *"With software there are only two possibilites: either the users control the programme or the programme controls the users. If the programme controls the users, and the developer controls the programme, then the programme is an instrument of unjust power."*
>
> Richard Stallman

Fork

Diseñe e implemente un programa que dado un proceso padre, cree N procesos hijos del mismo proceso padre usando la syscall **fork**.

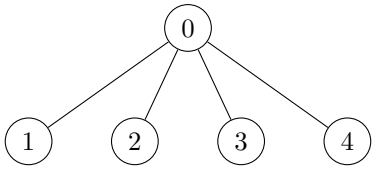

Figura 1: Ejemplo del árbol pedido en el enunciado, con N = 4

Entrada

- Un entero N, que representa la cantidad de hijos a crear.

Salida

- El pid del proceso principal, seguido del pid y ppid de los N procesos creados.

Caso de prueba

- **Input:**
  3
- **Output:**
  PROCESO PRINCIPAL (PID 19668)
  HIJO 1 (PID 19669) DEL PADRE (PID 19668)
  HIJO 2 (PID 19670) DEL PADRE (PID 19668)
  HIJO 3 (PID 19671) DEL PADRE (PID 19668)

"Time-travel is possible, but no person will ever manage to kill his past self. Godel laughed his laugh then, and concluded, The a priori is greatly neglected. Logic is very powerful."

Kurt Gödel

Pipe

Diseñe un programa en C usando pipes en el cual uno de los procesos envía un mensaje(string) a un segundo proceso; al recibir el mensaje el segundo proceso replica al primero enviándole un mensaje de vuelta, donde las mayúsculas del mensaje recibido se cambian a minúsculas y las minúsculas se cambian a mayúsculas. (Omita símbolos de acentuación, diéresis, entre otros para simplificar el problema.)

Entrada:

- Un string s, que contiene el mensaje del proceso 1.

Salida:

- Un string z, que corresponde al mensaje s pero con las mayúsculas y minúsculas invertidas.

Caso de prueba

- **Entrada:**
  HolaaaAA
- **Salida:**
  hOLAAAaa

Gracias por su atención!