

MiniTP Shell, procesos e hilos

Leandro Nehemias Camperoz

27 de agosto de 2021

1. Shell y terminal.

Si bien era posible escribir un script con menos lineas de codigo, decidi usar una sintaxis un poco mas entendible. Usando variables mas descriptivas sobre los archivos y comandos con los que se esta trabajando.

```
#!/bin/bash

USUARIO=$(whoami)
NOMBRE_DIRECTORIO=$1
HOME_USUARIO=/home/$USUARIO/

if [ $NOMBRE_DIRECTORIO ];then
    # Las siguientes lineas se ejecutan si se paso un parametro
    # NOMBRE_DIRECTORIO != ""
    NUEVO_DIRECTORIO=$HOME_USUARIO/$NOMBRE_DIRECTORIO
    NUEVO_ARCHIVO=$NUEVO_DIRECTORIO/contenido_home.txt

    mkdir $NUEVO_DIRECTORIO && touch $NUEVO_ARCHIVO
    # Volcamos la salida del comando ls en contenido_home.txt
    # [-l muestra permisos, tamano, usuario, fecha y hora de creacion]
    # [-a no ignora archivos ocultos] [-v ordena por version ]
    # [--group-directories-first directorios primero]
    ls -lav --group-directories-first $HOME_USUARIO >\
    $NUEVO_ARCHIVO
else
    echo "Error:Debe ejecutar con un parametro por lo menos."
    exit 1
fi

cat $NUEVO_ARCHIVO
# Comando read
# Solicitamos al usuario que presione una teclas o ingrese algun parametro
# [-s no muestra caracteres ingresados ] [ -p muestra en pantalla mensaje ]
# [-r barra invertida no escapa caracteres ]
# Al leer la barra invertida como parte de lo que ingresa el usuario
# solo termina de leer por teclado cuando presionamos [Enter].
read -rsp $"Presione Enter para salir...\n"
exit 0
```

2. Estados de un proceso.

Despues de varios intentos logre capturar el cambio de estado del proceso. El el codigo fuente del programa usado esta en proceso.c incluido en el archivo zip enviado.

Programa esperando input del usuario.

```
0[|] 1.3% Tasks: 57, 289 thr; 1 running
1[|] 0.7% Load average: 0.24 0.99 1.42
2[|] 7.8% Uptime: 02:53:10
3[|] 2.0%
Mem[|||||] 1.44G/7.11G
Swp[|] 0K/2.00G

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
5500 nehemias 20 0 2364 680 596 S 0.0 0.0 0:00.00 ./proceso

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice +F9Kill F10Quit
```

Programa realizando operaciones aritmeticas y mostrando por pantalla el resulatdo cada 1 microsegundo.

```
0[|||||] 15.6% Tasks: 57, 287 thr; 2 running
1[|||||] 15.6% Load average: 0.63 0.94 1.37
2[|||||] 15.0% Uptime: 02:54:10
3[|||||] 17.4%
Mem[|||||] 1.38G/7.11G
Swp[|] 0K/2.00G

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
5500 nehemias 20 0 2364 680 596 R 12.2 0.0 0:00.64 ./proceso

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice +F9Kill F10Quit
```

3. Threads.

Luego de modificar el codigo suministrado en el enunciado del trabajo practico y ejecutar el programa. Obtuve como resultado una diferencia considerable en el tiempo de ejecucion. El codigo del programa utilizado esta en el archivo hilos.c.

Prueba realizada en un procesador intel i3 dual-core, con 2 LCPU por core, 2.3Ghz de frecuencia maxima.

```
~/Documentos/Sor/miniTps/miniTp1 > ./hilos
Prueba de ejecucion sin pthread: 26 segundos
Prueba de ejecucion con pthread: 8 segundos
Diferencia de 18 segundos, una mejora del 69% en tiempo de ejecucion.
~/Documentos/Sor/miniTps/miniTp1 >
```

Captura htop, ejecucion del programa sin pthread.

```
0[| 0.7%] Tasks: 58, 307 thr; 2 running
1[| 0.6%] Load average: 0.72 0.63 0.65
2[| 0.7%] Uptime: 03:21:55
3[| 100.0%]
Mem[| 1.456/7.115]
Swp[| 0K/2.000]

  PID USER   PRI  NI  VIRT   RES   SHR S %CPU  %MEM   TIME+ Command
 6531 nehemias  20    0  2368    776   688 R  99.5   0.0  0:02.83 ./hilos

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

Captura htop, ejecucion del programa con pthread.

```
0[| 98.7%] Tasks: 58, 312 thr; 4 running
1[| 98.7%] Load average: 1.08 0.62 0.65
2[| 99.3%] Uptime: 03:20:39
3[| 100.0%]
Mem[| 1.446/7.115]
Swp[| 0K/2.000]

  PID USER   PRI  NI  VIRT   RES   SHR S %CPU  %MEM   TIME+ Command
 6505 nehemias  20    0  43480    776   688 R  88.1   0.0  0:01.33 ./hilos
 6506 nehemias  20    0  43480    776   688 R  73.5   0.0  0:01.11 ./hilos
 6507 nehemias  20    0  43480    776   688 R  71.5   0.0  0:01.08 ./hilos
 6508 nehemias  20    0  43480    776   688 R  79.5   0.0  0:01.20 ./hilos
 6509 nehemias  20    0  43480    776   688 R  81.5   0.0  0:01.23 ./hilos
 6494 nehemias  20    0  43480    776   688 S  396.   0.0  0:32.13 ./hilos

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```