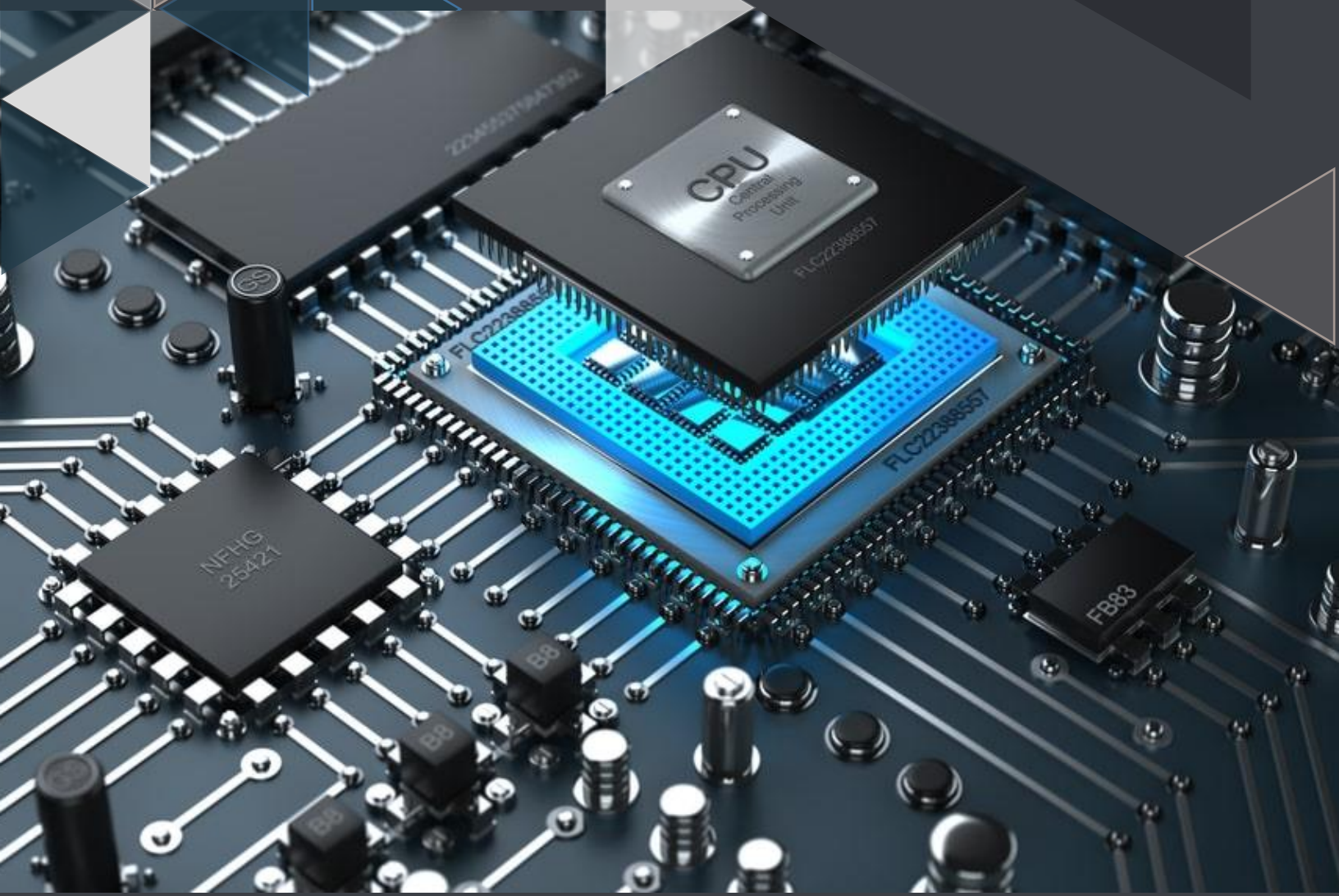
HILOS,SEÑALES

Ing. Nelson Belloso

CLASE 04

Sistemas Operativos
SIO104



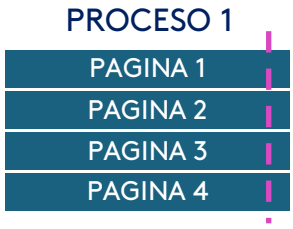
AGENDA

Hilos

Señales

THREADS HILOS

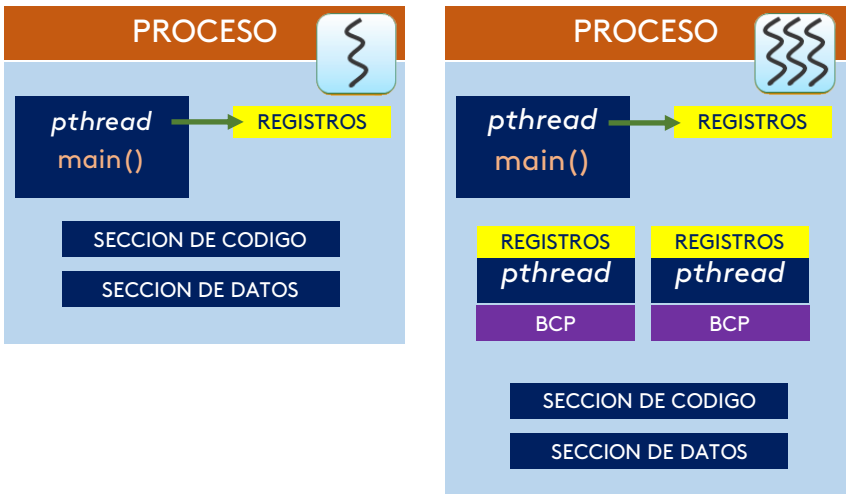
Un programa es un proceso, cuando esta cargado en memoria. Se divide en varias paginas que contienen bloques de sentencias listas para ejecutarse.



Siendo un proceso la conformación de varias paginas de instrucciones.

Hilo del proceso representa la ejecución ordenada en secuencia de todas las instrucciones de cada pagina, hasta el final de la ejecución del proceso.

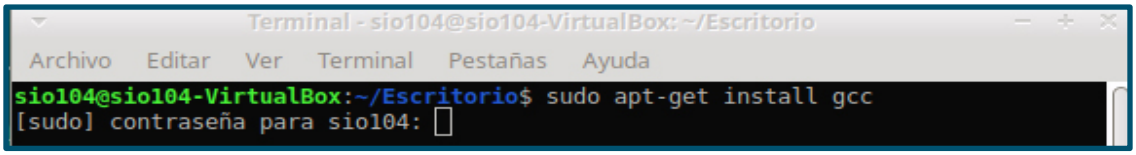
- Un proceso normal inicia ejecutando su punto de entrada (**main()**)como un único hilo.
- Un hilo puede crear otros hilos dentro del mismo proceso y cada uno tendrá su propio **stack de registros** (Contador de programa, pila, registros)

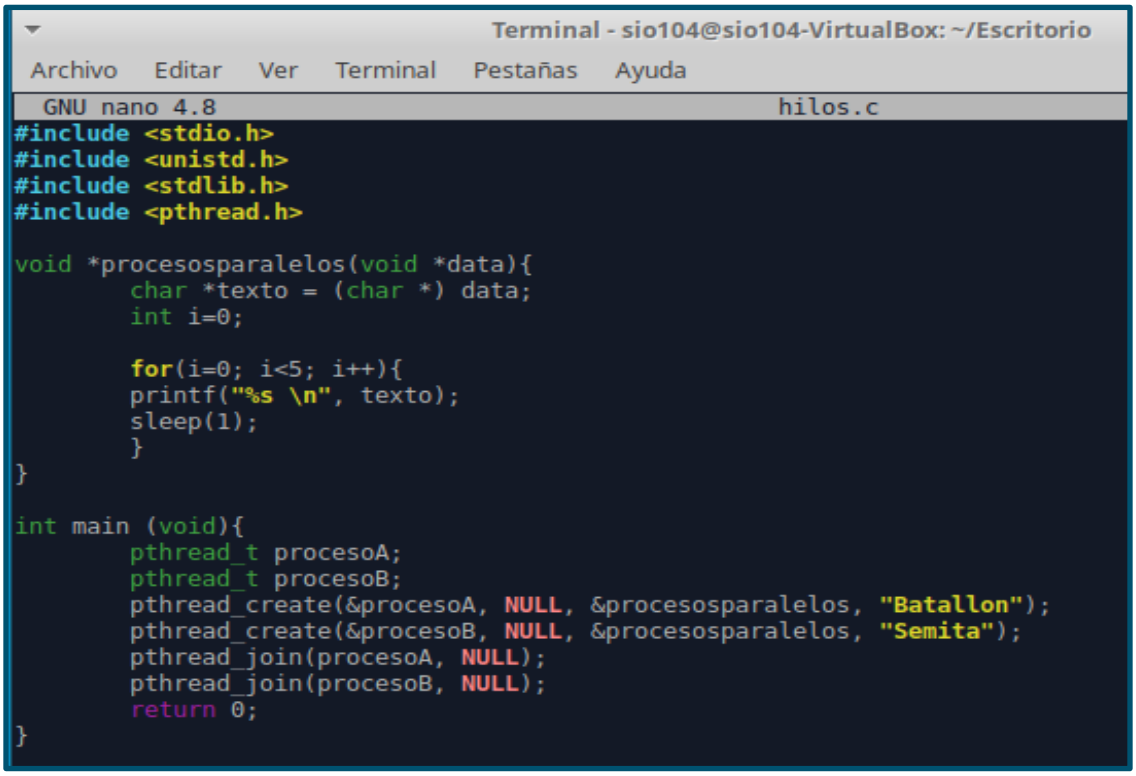


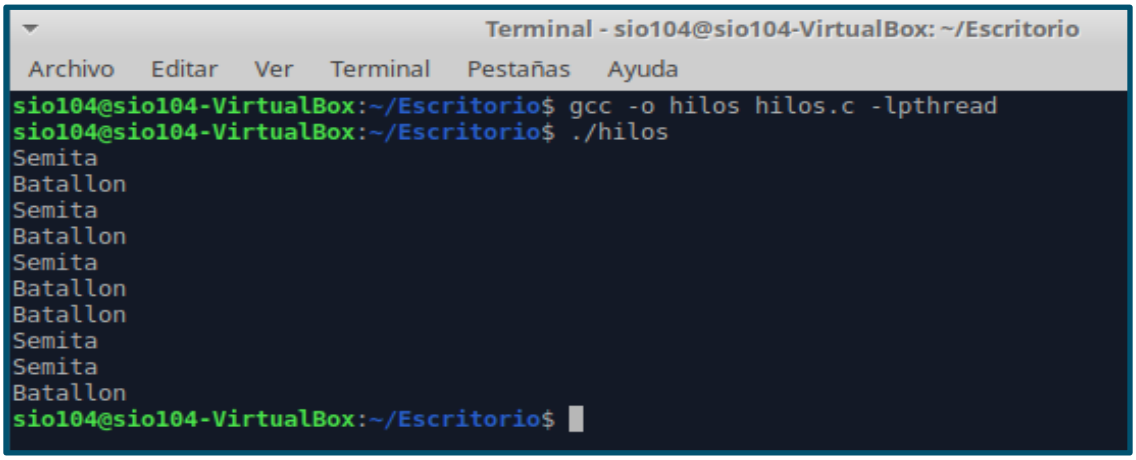
Los hilos pertenecientes a un mismo proceso, comparten la información de este proceso en común.

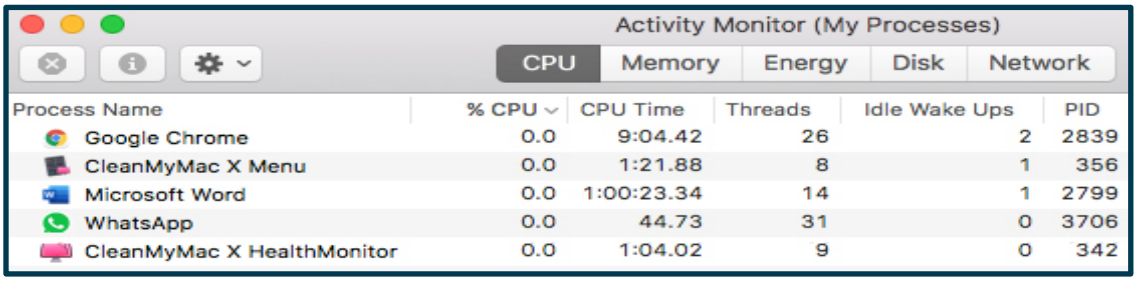
- Espacio de memoria del proceso
- Variables globales
- Ficheros abiertos
- Procesos hijos

Instalación de paquetes y programas







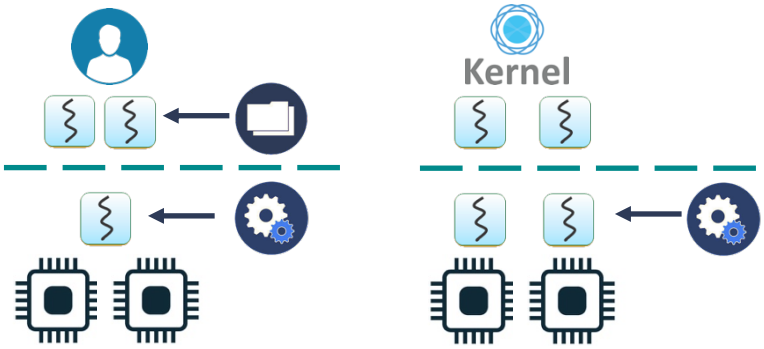


Gestión de hilos

Existen hilos a nivel de usuarios y también existen hilos a nivel de **kernel** cada tipo toma vida en sus espacios y tienen su propio planificador.

- ULT User level Thread Gestionados totalmente por la biblioteca (panificador de hilos). **El sistema Operativo no tiene conocimiento de la existencia de estos Hilos.**

- KLT Kernel Level Thread Son gestionados por el planificador de proceso del sistema Operativo



La existencia de los hilos permite incrementar el rendimiento de los procesadores, puesto que procesos complejos pueden simplificarse dividiéndolos en múltiples procesos simples (**múltiples hilos**).
Los hilos fueron concebidos para realizar tareas lentas las cuales no deben paralizar el programa principal y/o proceso principal Ejemplo:

- Grabación-almacenamiento de información en disco
- Conexión a un servidor

SEÑALES Y EXCEPCIONES

Los sistemas operativos pueden comunicarse con los procesos en ejecución mediante **señales** (notificación enviada a un proceso determinado) para el caso de los sistemas **Windows** se utilizan **excepciones.**

Señales UNIX. Las señales se utilizan para notificar a un proceso sobre una solicitud para realizar un evento. Las señales funcionan con los procesos de una manera muy similar al funcionamiento de el procesador con las interrupciones.

Haciendo uso del comando **kill -l** podemos observar las diferentes opciones con las que cuenta la instrucción.

```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
 GNU nano 4.8                        señal.c
#include <stdio.h>
#include <unistd.h>
#include <signal.h>

void mysenal(int sig)
{
        printf("numero de señal %d \n", sig);
}

int main(void){
        signal(SIGIO, &mysenal);
        while(1)
        {
        sleep(2);
        printf("llinas a dolar \n");
        }
return 0;
}
                        [ 18 líneas leídas ]
^G Ver ayuda ^O Guardar   ^W Buscar     ^K Cortar Tex^J Justificar^C Posición
^X Salir     ^R Leer fich.^\ Reemplazar^U Pegar      ^T Ortografía^_ Ir a línea
```

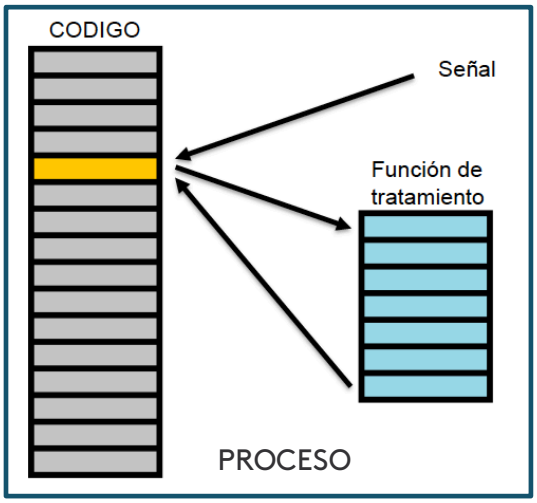
```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
sio104@sio104-VirtualBox:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+1
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-1
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3
63) SIGRTMAX-1  64) SIGRTMAX
sio104@sio104-VirtualBox:~$ ps u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sio104      1110  0.0  0.2  21268  5236 pts/0    Ss   15:20   0:00 bash
sio104      2807  0.0  0.2  21096  5032 pts/1    Ss   16:29   0:00 bash
sio104      2834  0.0  0.0   2496   584 pts/0    S+   16:32   0:00 ./señal
sio104      2845  0.0  0.1  21920  3416 pts/1    R+   16:33   0:00 ps u
sio104@sio104-VirtualBox:~$ kill -18 2834
sio104@sio104-VirtualBox:~$ kill -29 2834
sio104@sio104-VirtualBox:~$ kill -9 2834
```

```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
llinas a dolar
llinas a dolar
llinas a dolar
llinas a dolar

[1]+  Detenido                 ./señal
sio104@sio104-VirtualBox:~$ llinas a dolar
llinas a dolar
llinas a dolar
llinas a dolar
llinas a dolar
llinas a dolar
numero de señal 29
llinas a dolar
```

Cuando un proceso recibe una señal se comporta de la siguiente manera

- El proceso detiene la ejecución de la instrucción maquina que se esta ejecutando
- Ejecuta una rutina de tratamiento a la señal.
- Una vez ejecutada la rutina de señal continua en el punto que interrumpió el proceso



Señal proceso -> proceso es cuando un proceso es el encargado de enviar una señal a otro proceso, pero solo si estos poseen el mismo identificador.

Señal SO -> Proceso Esto se da cuando el sistema operativo requiere que el flujo normal de instrucciones del proceso sea alterado para ejecutar nuevas instrucciones.

Excepciones Windows es un evento que ocurre durante la ejecución de un proceso y que requiere la ejecución de un fragmento de código situado fuera del flujo normal de ejecución.

Una excepción puede ser generada por

- Excepción de hardware detecta condiciones especiales, desbordamiento de buffer, violación de segmento, fallo de pagina solo lectura.
- Un proceso
- Sistema operativo. Salva el contexto del proceso, envía la excepción al proceso y ejecuta la rutina.

El manejo de excepciones de realiza habitualmente a través de un lenguaje de programación por medio de bloques