



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Viernes, 09/02/2024
Hora de inicio:	15:40 - 17:20

Sistemas Operativos 1 “A”

Jhonathan Daniel Tocay Cotzoyay

Primer Semestre 2024

Sistemas Operativos 1

Jhonathan Daniel Tocay Cotzajay

Datos del Auxiliar



- Jhonathan Tocay
- Correo Electrónico:
2878571900109@ingenieria.usac.edu.gt
- Asunto: [SO1]Asunto
- Usuario GitHub: JhonathanTocay2020
- Repositorio:
[https://github.com/JhonathanTocay2020/Laboratorio SO1 1S2024.git](https://github.com/JhonathanTocay2020/Laboratorio_SO1_1S2024.git)

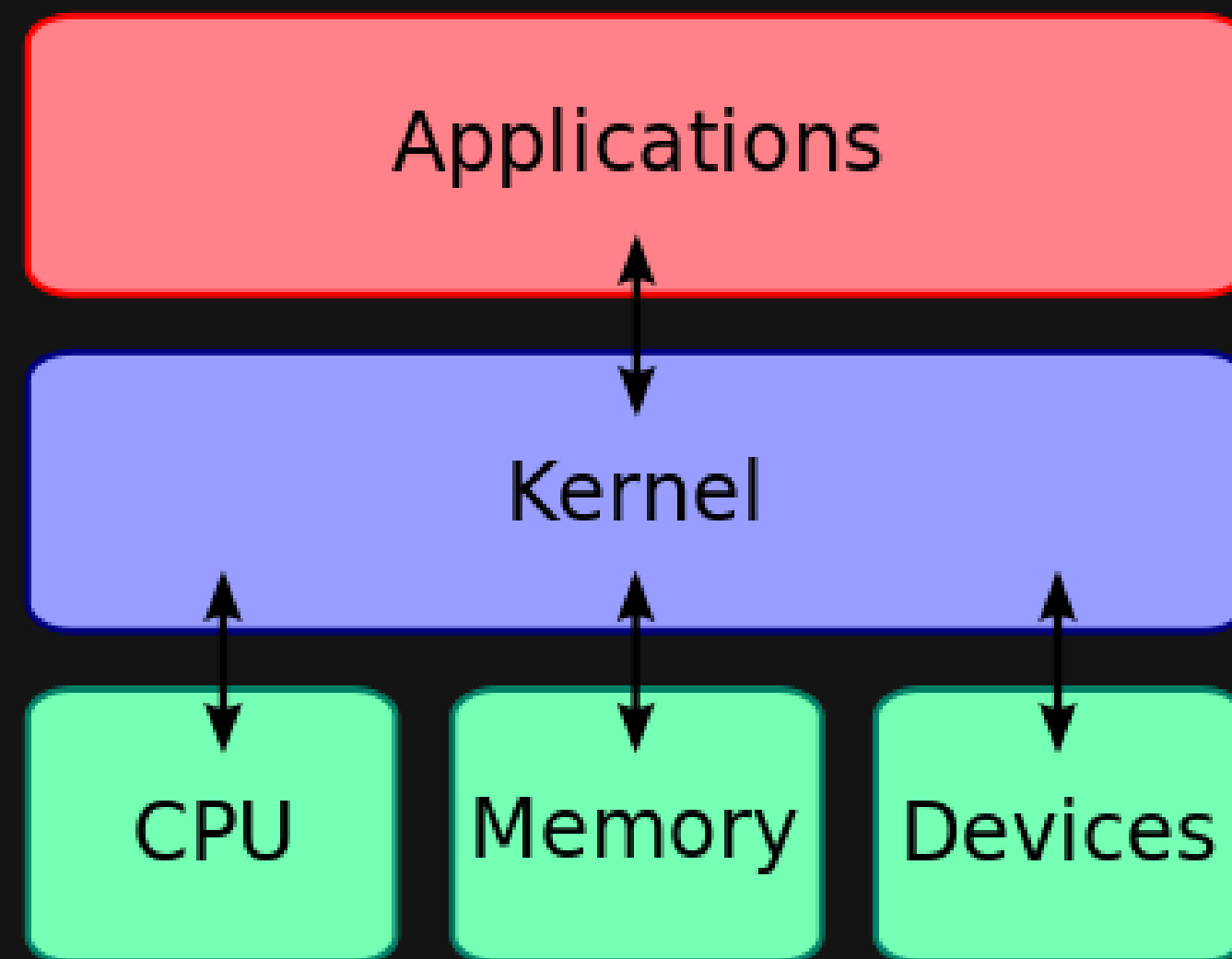
Agenda



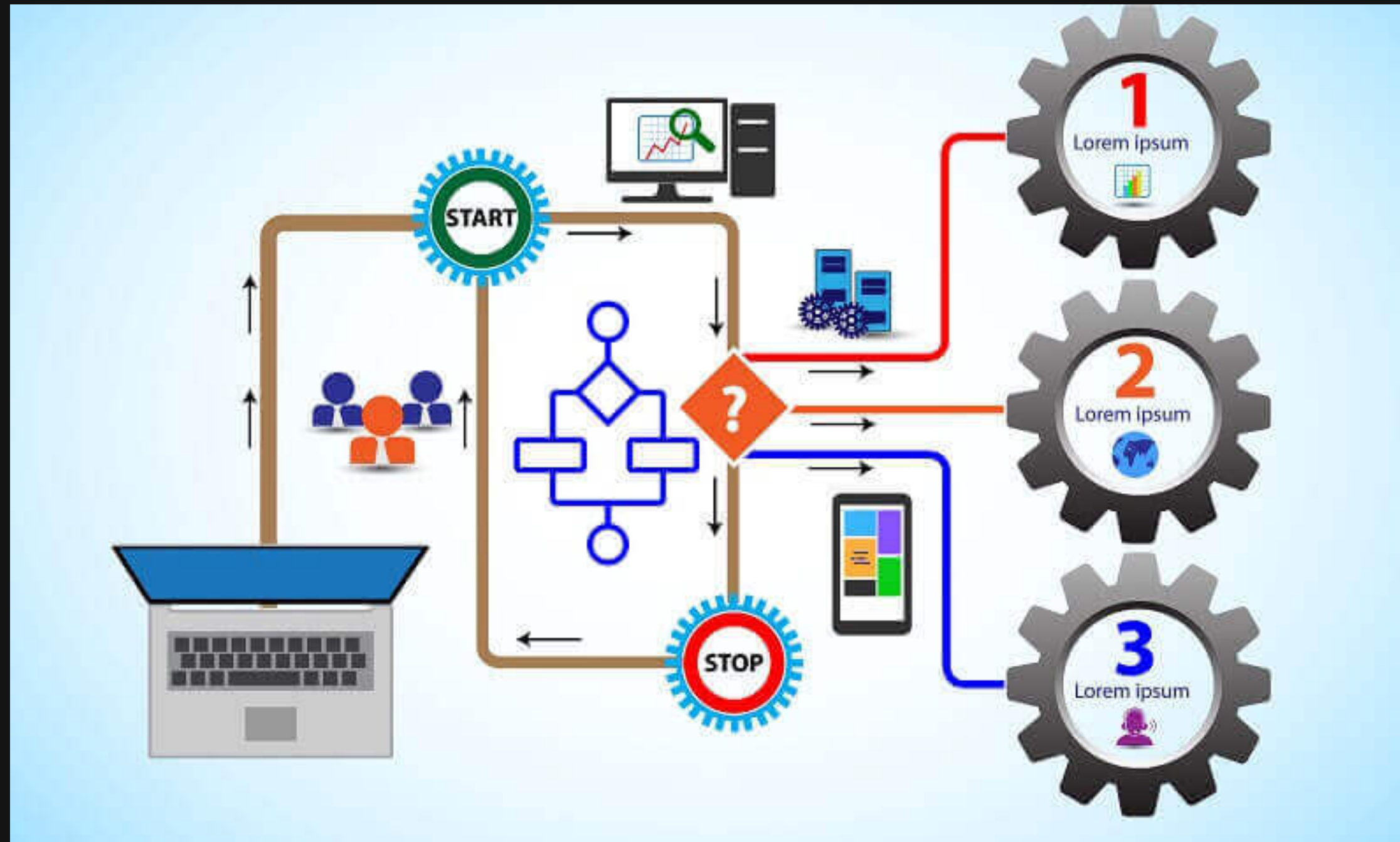
- **AVISOS**
- **Foro 3**
- **Modulos de Kernel**
- **Ejemplos Practicos**
- **Hoja de Trabajo 1**
- **Formulario de Asistencia**

Kernel

Kernel es un programa que da soporte al hardware de la computadora.



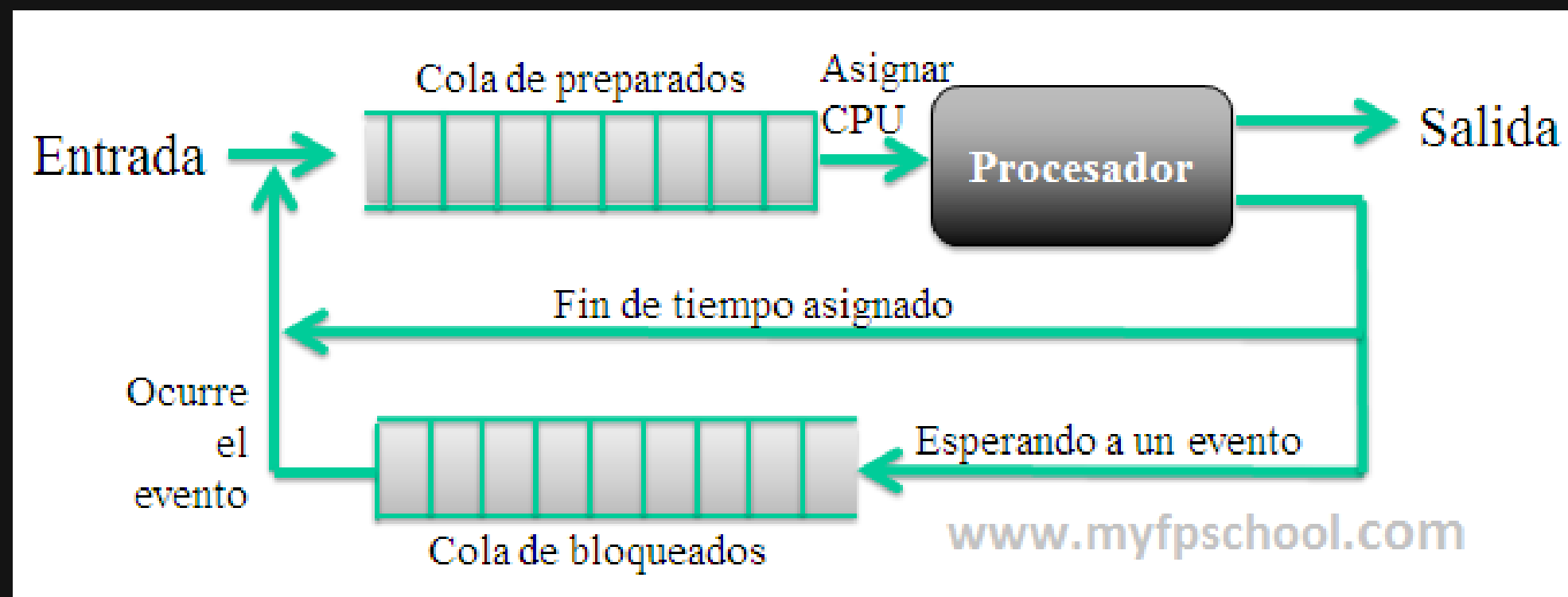
Estados de Procesos y Concurrency



Proceso

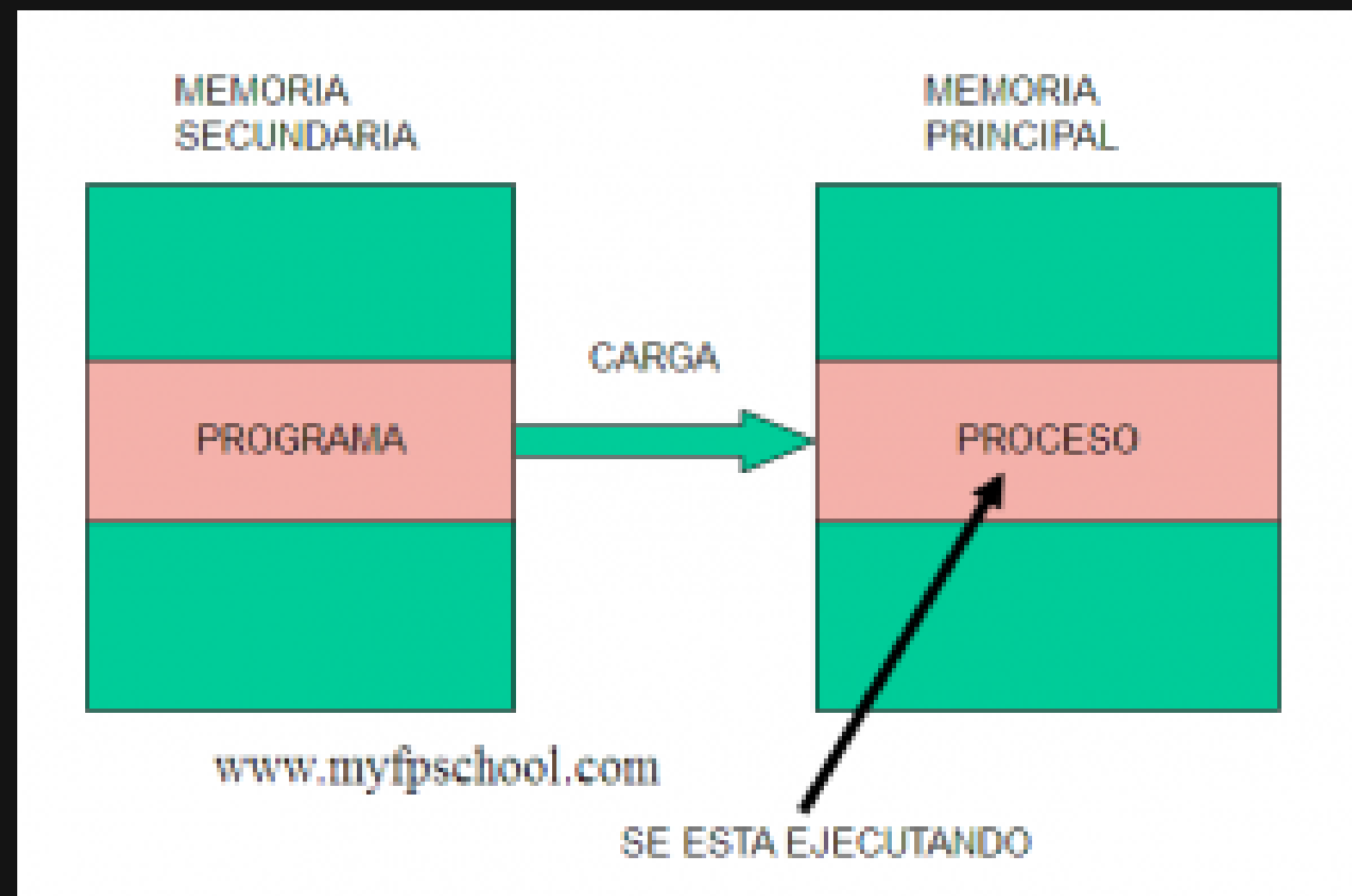
Un proceso es un programa en ejecución, este programa tiene varias partes, este tiene segmentos de memoria como Heap, Stack, Data y Text.

Los procesos manejan un diagrama de estados y cada proceso tiene información como por ejemplo: ID de procesos (PID), nombre (name), dueño (owner), segmentos de memoria, programa que está ejecutando y estado.

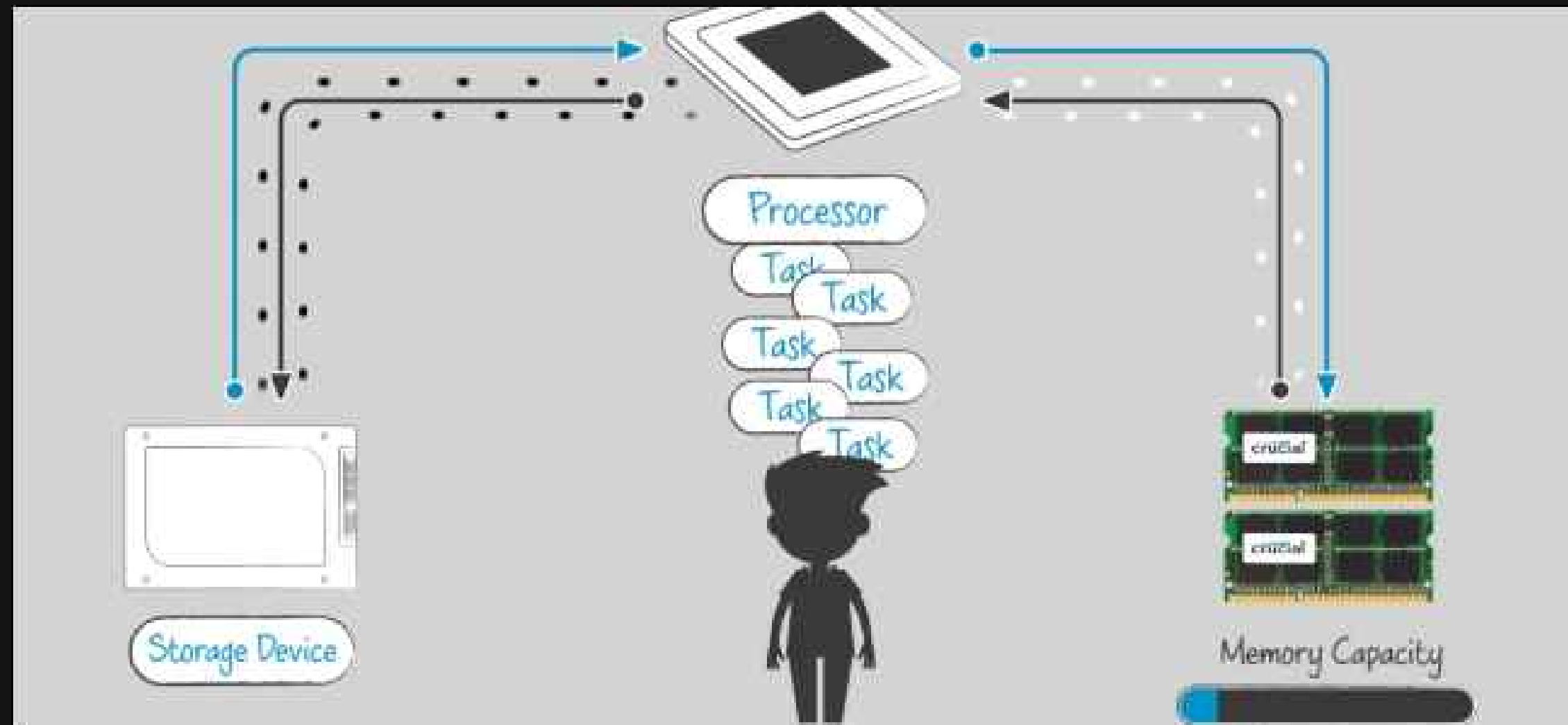


Procesos y Programas

Un programa reside en memoria secundaria generalmente y son una serie de órdenes que sirven para ejecutar una tarea con un fin determinado. Un proceso al contrario que un programa se ubica en memoria principal y se está ejecutando.



En memoria puede haber muchos procesos ejecutándose a la vez (de forma concurrente), por lo tanto van a ir demandando una serie de recursos (memoria, CPU, etc.), y será la CPU como ya se ha dicho la encargada de suministrar dichos recursos de forma ordenada.



Estados de los Procesos

New



- Cuando se crea un proceso, se almacena en el heap y se llama task.
 - Task: Es el segmento de memoria para el proceso, cuando éste procesa información se convierte en hilo.
 - Hilo: Unidad de flujo de procesamiento

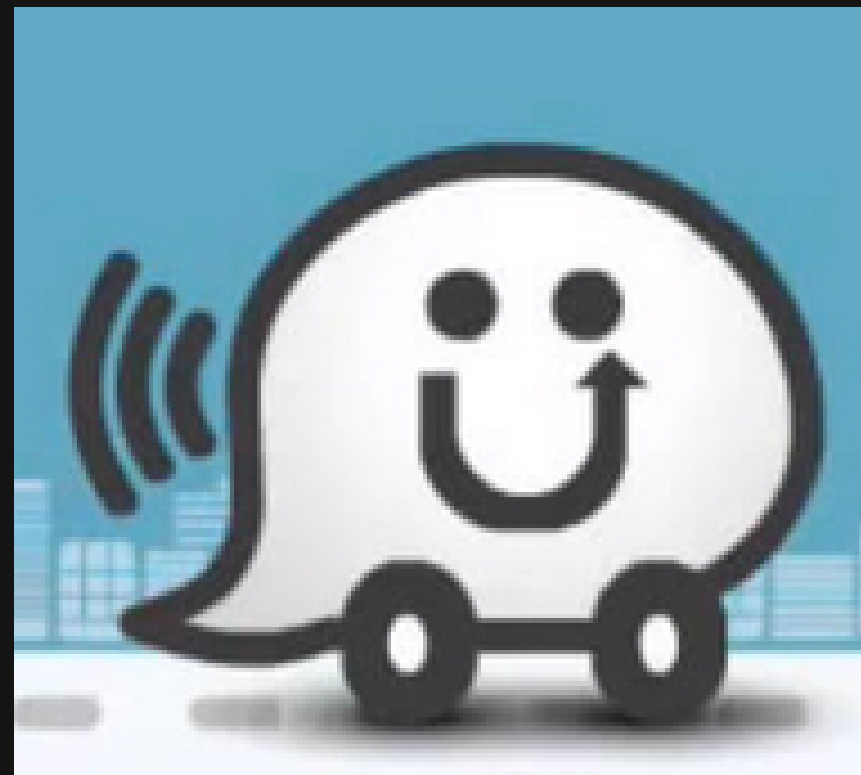
Ready

Aquí se manejan colas de procesos listos para ser ejecutados.



Running

Es cuando el proceso se ejecuta en el microprocesador. En este contexto está el reloj de la computadora que marca el proceso que estará un proceso en running y luego regresará a ready.



Waiting

Cuando un dispositivo está pidiendo acceso por hardware y cuando ya está listo, regresa a Ready.



Finished

Cuando termina un proceso.



Señalización de Procesos

El comando básico para enviar señales a un proceso es kill



Detalles

Ctrl-C y Ctrl-Z son atajos para enviar señales SIGINT (2) y SIGTSTP (20)

```
/ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=118 time=9.017 ms
64 bytes from 8.8.8.8: seq=1 ttl=118 time=9.074 ms
64 bytes from 8.8.8.8: seq=2 ttl=118 time=8.929 ms
^C ← Ctrl + C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 8.929/9.006/9.074 ms
/ # ← Ctrl + D
marko@pnap:~$
```

Sintaxis de Kill

kill [señal] PID

- **SIGHUP (1):** cuelgue del terminal o muerte del proceso controlador
- **SIGTERM (15):** mata el proceso permitiéndole terminar correctamente
- **SIGKILL (9):** mata el proceso sin permitirle terminar
- **SIGSTOP (19):** para el proceso
- **SIGCONT (18):** continúa si parado
- **SIGINT (2):** interrupción de teclado (Ctrl-C)
- **SIGTSTP (20):** stop de teclado (Ctrl-Z)
- **SIGQUIT (3):** salida de teclado (Ctrl-\)

Características

- La señal que se envía por defecto es TERM (15)
 - los procesos pueden ignorar esta señal y no terminar
 - las señales KILL (9) y STOP (19) no pueden ignorarse
- En bash, cuando enviamos una señal SIGHUP a un shell, este se lo reenvía a todos sus hijos
- Cuando cerramos un terminal en un entorno gráfico, o abandonamos una sesión, se envía un SIGHUP a todos sus hijos
- La mayoría de los demonios (daemons) responden a la señal SIGHUP volviendo a leer sus ficheros de configuración:
 - en vez de matar y reiniciar un demonio podemos hacer un kill -HUP para reiniciarlo

Otros Comandos

- **pgrep funciona similar a ps | aux**
- **pkill envía señales a procesos por nombre**

Directorio /proc

Contiene un sistema de archivos imaginario o virtual.

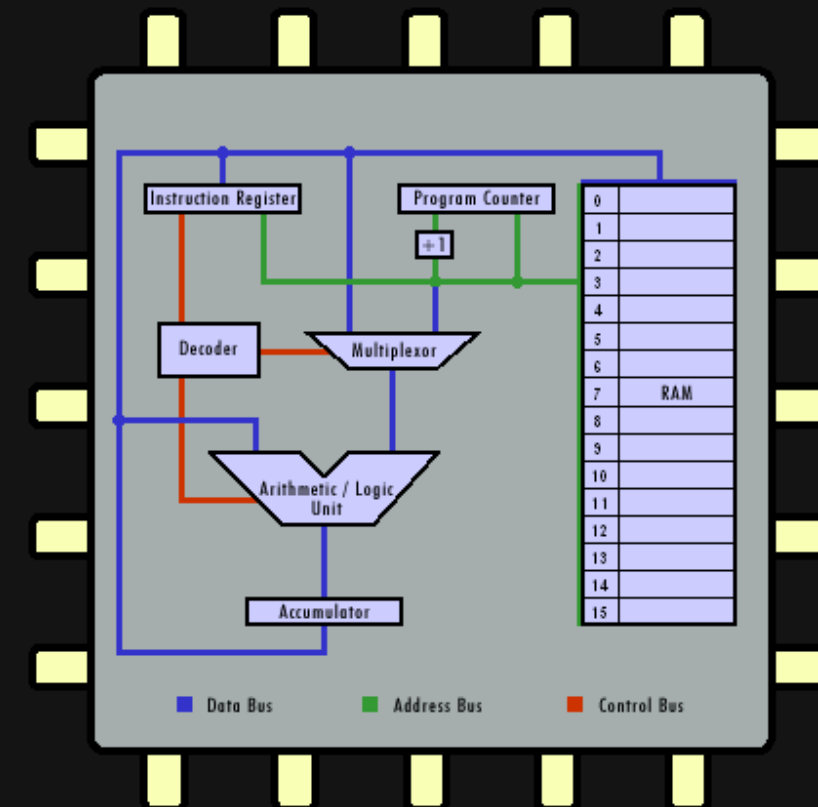
Este no existe físicamente en disco, sino que el núcleo lo crea en memoria.

Se utiliza para ofrecer información relacionada con el sistema.

Archivos importantes de /proc

/proc/cpuinfo:

Información acerca del procesador: su tipo, marca, modelo, rendimiento, etc.



/proc/devices

Lista de controladores de dispositivos configurados dentro del núcleo que está en ejecución.

/proc/ioproports

Información de los puertos de E/S que se estén utilizando en cada momento.

/proc/meminfo

**Información acerca de la utilización de la memoria
física y del archivo de intercambio.**

kernel.h

**Librería de C usada para conocer información del Kernel
directamente de los módulos en el lenguaje de programación**

Manejo de Kernel en Docker

- Montar carpeta /proc
- Dar privilegios de super usuario
- Obtener PIDs del host

Host

