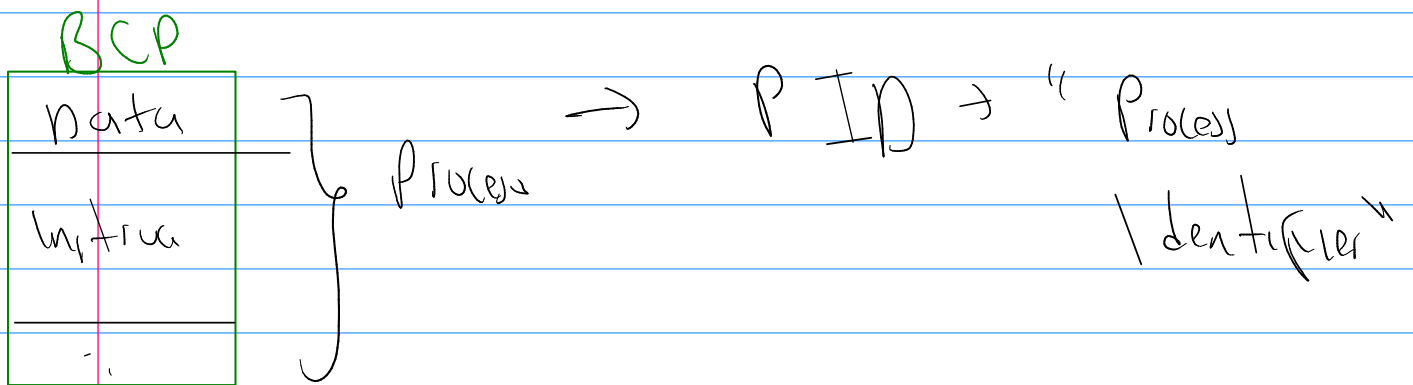


Proceso

↳ Programa en ejecución (que se está ejecutando o ya se ejecutó)

↳ Se ejecuta en memoria, para ello hace una copia del archivo ejecutable en memoria. Armandolo la estructura del programa (BCP) → "Process control block"



• (ps -e) → lista de procesos activos

Cambiando un apuntador de un PID

basta por que el so ejecute otro archivo

o programa

Ejemplo de process.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command	Uptime
35573	user	20	0	2674M	182M	103M	S	1.3	1.1	2:40.83	/snap/firefox/2487/usr/lib/firefox/firefox -co	
1239	user	20	0	24.8G	89376	41116	S	0.7	0.6	1:55.44	/usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run	
4205	user	20	0	12.7G	741M	236M	S	0.7	4.7	0:29.00	/snap/firefox/2487/usr/lib/firefox/firefox	
44560	user	20	0	33.2G	154M	82464	S	0.7	1.0	0:00.46	/usr/share/code/code --type=gpu-process --enab	
49316	user	20	0	800M	53376	40752	S	0.7	0.3	0:00.59	/usr/libexec/gnome-terminal-server	
49367	root	20	0	15656	6824	3688	R	0.7	0.0	0:00.05	htop	
1	root	20	0	164M	13312	7936	S	0.0	0.1	0:05.83	/sbin/init splash	

↳ `ps -e`

PID ⇒ Process Identifier

PRI ⇒ Priority (Prioridad real del proceso)

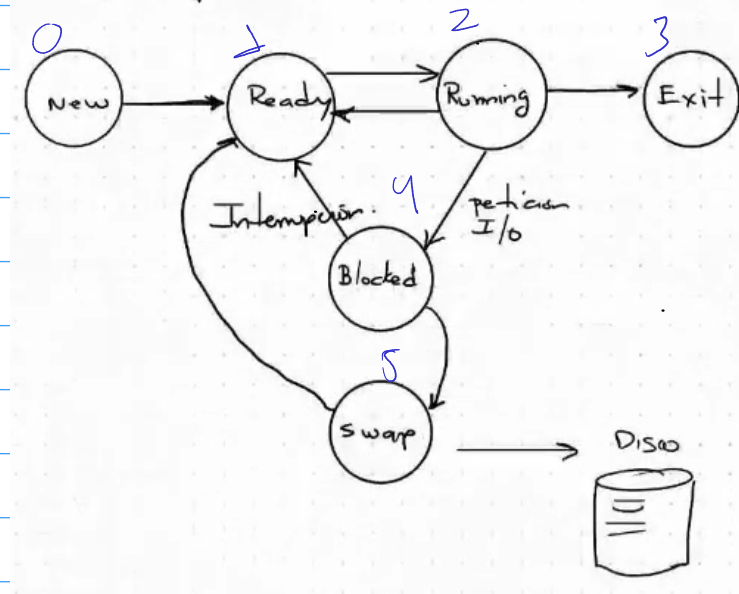
NI ⇒ Nice (Prioridad de corteja)

La diferencia entre PRI y NI es que NI es la prioridad que nosotros como usuarios asignamos y que ajustamos de acuerdo a nuestras necesidades.

PRI se calcula como $PRI = 20 + NI$, y para evitar que un proceso ocupe todo el procesor del computador se tiene un counter que disminuye con el paso del tiempo para evitar esto.

Estados de un proceso:

Estados de un proceso



Hay 6 estados para un proceso.

→ 0 (new) ⇒ Es la creación de un proceso

→ 1 (Ready) ⇒ Se usa para determinar que un proceso está listo para ejecutar

→ 2 (Running) ⇒ Proceso que se encuentra en ejecución

→ 3 (Exit) ⇒ Proceso que ya finalizó

→ γ (Blocked) →

Proceso que se pauso por una causa especifica, I/O es decir, que esta esperando una entrada o salida de datos, por que el usuario lo aborto o por que hubo un error

→ Swap (6) →

Se usa en ocasiones la memoria swap para que un proceso se ejecute en este tipo de memoria para no ocupar la principal y evitar así una sobrecarga.

Dependiendo del uso que se le de al computador se puede aumentar este tipo de memoria, sabiendo siempre que así no se este usando la memoria swap siempre se tendrá ese espacio destinado, es decir, que no estará libre

Program Counter

↳ El program counter de un SO es aquel puntero que le indica a la CPU que proceso es el siguiente en ejecutarse. Cada vez que se enciende el computador el program counter inicia en 0 y se va a aumentando. Es importante por que evita que la CPU no sepa que instrucción se va a ejecutar enseguida.

También es usado para subrutinas, es decir, que cuando se tiene una subrutina el program counter apunta a la nueva rutina y esta una vez finaliza el program counter apunta nuevamente a la rutina principal

Creación de procesos

↳ fork()

↳ Comunicación de procesos

↳ Tipos de procesos

↳ Primer plano

↳ Daemon

↳ Zombies