**Este comando monitoriza en tiempo real el proceso llamado cpu\_killer, actualizándose cada segundo**

*watch -n 1 “ps -eo pid,comm,cls,pri,ni,%cpu,time,etime,psr - -sort=-%cpu | grep cpu\_killer”*

**watch -n 1:** Ejecuta el comando cada 1 segundo.

**ps -eo ...:** Muestra información sobre los procesos.

-**-sort**=-%cpu: Ordena los procesos por uso de CPU en orden descendente.

**grep cpu\_killer:** Filtra para mostrar solo el proceso cpu\_killer

**Muestra la información de todos los procesos, pero solo devuelve la información sobre cpu\_killer**

*ps -eo pid,comm,cls,pri,ni,%cpu,time,etime,psr - -sort=-%cpu | grep cpu\_killer*

**ps -eo ...:** Igual que antes.

**--sort=-%cpu:** Ordena por uso de CPU.

**grep cpu\_killer:** Filtra la salida para el proceso específico.

**Lanza múltiples instancias del proceso cpu\_killer, cada una con diferentes configuraciones de prioridad**

*(sudo taskset -c 0 ./cpu\_killer &) && (sudo taskset -c 0 nice -n 19 ./cpu\_killer &) && (sudo taskset -c 0 nice -n -20 ./cpu\_killer &)*

**taskset -c 0:** Restringe la ejecución al núcleo 0.

**nice -n 19:** Ejecuta con la prioridad más baja.

**nice -n -20:** Ejecuta con la prioridad más alta.

**&:** Ejecuta en segundo plano.

**Ejecuta cpu\_killer con diferentes prioridades fijas en tiempo real**

*(sudo taskset -c 0 chrt -f 1 ./cpu\_killer &) && (sudo taskset -c 0 chrt -f 20 ./cpu\_killer &) && (sudo taskset -c 0 chrt -f 25 ./cpu\_killer &)*

**chrt -f N:** Establece la prioridad en tiempo real (fijo). N puede ser entre 1 y 99 (donde 1 es mayor prioridad).

Se ejecutan tres instancias con prioridades 1, 20 y 25, todas en el núcleo 0.

**Ejecuta cpu\_killer con diferentes prioridades dinámicas en tiempo real**

*(sudo taskset -c 0 chrt -r 1 ./cpu\_killer &) && (sudo taskset -c 0 chrt -r 20 ./cpu\_killer &) && (sudo taskset -c 0 chrt -r 25 ./cpu\_killer &)*

**chrt -r N:** Cambia la política de planificación a tiempo real con prioridad dinámica. La prioridad puede cambiar durante la ejecución dependiendo de la carga del sistema.

Similar al comando anterior, pero aquí se utilizan prioridades dinámicas

**Lanza tres instancias del proceso cpu\_killer con la misma prioridad dinámica**

*(sudo taskset -c 0 chrt -r 50 ./cpu\_killer &) && (sudo taskset -c 0 chrt -r 50 ./cpu\_killer &) && (sudo taskset -c 0 chrt -r 50 ./cpu\_killer &)*

Aquí se utiliza chrt -r 50 para establecer la prioridad a 50 para cada instancia, lo que permite estudiar cómo se comportan múltiples procesos bajo la misma carga de prioridad

**Muestra el tiempo de fragmentación (en milisegundos) que se asigna a los procesos en tiempo real bajo la política de Round Robin (RR).**

*cat /proc/sys/kernel/sched\_rr\_timeslice\_ms*

**/proc/sys/kernel/sched\_rr\_timeslice\_ms:** Este archivo contiene el tiempo que los procesos en tiempo real pueden ejecutarse antes de ser interrumpidos. Un valor menor significa que los procesos se intercambiarán más a menudo, permitiendo una mayor reactividad.

**Establece el tiempo de fragmentación para los procesos en tiempo real a 500 milisegundos.**

*cat /proc/sys/kernel/sched\_rr\_timeslice\_ms=500*

**echo 500 > /proc/sys/kernel/sched\_rr\_timeslice\_ms:** Este comando utiliza echo para escribir un nuevo valor en el archivo de configuración del sistema, cambiando el tiempo que un proceso puede ejecutarse antes de ser interrumpido. Al hacerlo, se puede ajustar el comportamiento del planificador de CPU para procesos en tiempo real

**DIA 4 OCTUBRE**

sudo vim /etc/default/grub → poner Menu, 10 y UBUNTU

update-grub2

en lubuntu pulsamos el botón E y en línea donde ponga init.rd la borramos y pulsamos F10

grub-mkpasswd-pbkdf2

sudo cpio -idmv < /boot/initrd.img-6.8.0-40-generic

ls -lat

gedit & →zona de memoria ocupada por el proceso

para saber cual es el código nos fijamos en la X de ejecución

**8 DE OCTUBRE**

El heap es la zona de memoria que guarda el programador, lo diferenciamos con el comando asociado: rw-p

**Cosas interesantes del directorio proc:**

* Para mostrar información de la CPU:
  + cpu info
* Para mostrar sistema de memoria
  + free -h
* Para mostrar la prioridad de los ficheros
  + cat /proc/swaps
* Para mostrar
* cat /proc/ultimate
* Para mostrar
* cat /proc/loadavg
* Para mostrar las interrupciones de los programas
  + cat /proc/interrupts
* Para mostrar todos los modulos cargados en el kernel
* Para mostrar
* cat /proc/modules | grep ahci
* Para mostrar
* Para mostrar
  + ls -lart /sys/
* Para mostrar
  + ls -lart /sys/dev/block/
* Para mostrar
  + ls -lart /sys/dev/char
* Para mostrar
  + ls -lart /sys/dev/block/8:1