Multiprocesadores

Explicación de práctica 6

Sistemas Operativos

Facultad de Informática
Universidad Nacional de La Plata

2025

SMP en Linux

- Sockets: CPUs físicas en un package (chip) individual.
- Cores: Núcleos en una CPU física.
- Hyper Threading: Capacidad de paralelizar ciertas tareas dentro de un core sin implementar cores completos.
- Cores lógicos:
  - Sockets * Cores Físicos * HyperThreads por Core

lscpu

```
$ lscpu
Arquitectura:                         x86_64
  modo(s) de operación de las CPUs:   32-bit, 64-bit
  Address sizes:                      39 bits physical, 48 bits virtual
  Orden de los bytes:                 Little Endian
CPU(s):                               12
  Lista de la(s) CPU(s) en línea:     0-11
ID de fabricante:                     GenuineIntel
  Nombre del modelo:                  12th Gen Intel(R) Core(TM) i5-1235U
    Familia de CPU:                   6
    Modelo:                           154
    Hilo(s) de procesamiento por núcleo: 2
    Núcleo(s) por «socket»:           10
    «Socket(s)»                       1
```

} Logical Cores

Hyperthreading

Cores físicos

Sockets/Packages

cat /proc/cpuinfo

```
$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 154
model name      : 12th Gen Intel(R) Core(TM) i5-1235U
stepping        : 4
microcode       : 0x437
cpu MHz         : 1400.029
cache size      : 12288 KB
physical id     : 0
siblings        : 12
core id         : 0
cpu cores       : 10
```

Logical core id

Sockets/Packages

Physical core id

Physical cores

cat /proc/cpuinfo

```
$ cat /proc/cpuinfo | grep "core id"
core id         : 0
core id         : 0
core id         : 4
core id         : 4
core id         : 8
core id         : 9
core id         : 10
core id         : 11
core id         : 12
core id         : 13
core id         : 14
core id         : 15
```

Elementos fundamentales

Colección de productos	Procesadores Intel® Core™ i5 de 12ª Generación
Nombre de código	Productos anteriormente Alder Lake
Segmento vertical	Mobile
Número de procesador	i5-1235U
Litografía	Intel 7

Especificaciones de la CPU

Cantidad de núcleos	10
Cantidad de Performance-cores	2
Cantidad de Efficient-cores	8
Total de subprocesos	12
Frecuencia turbo máxima	4.40 GHz
Frecuencia turbo máxima del Performance-core	4.40 GHz
Frecuencia turbo máxima de Efficient-core	3.30 GHz
Caché	12 MB Intel® Smart Cache
Potencia base del procesador	15 W
Potencia turbo máxima	55 W
Potencia mínima asegurada	12 W

CPUs modernas

- SMP pero no todos los cores son iguales.
- Frequency scaling: El kernel puede cambiar la frecuencia de CPU para ahorrar energía.
- ARM BigLittle: Cores de bajo consumo y cores de alto consumo
- Intel Performance/Efficient cores: Similar a BigLittle
- Algunos cores pueden tener HyperThreading y otros no (en el ejemplo anterior solo 2 cores tienen hyperthreading, los que Intel llama “Performance Cores”).

Información de las CPUs

- /proc/cpuinfo
- lscpu
- Topología de la CPU: /sys/devices/system/cpu/cpu*/topology/
- Frequency scaling: /sys/devices/system/cpu/cpu*/cpufreq/

El scheduler

- El scheduler de Linux tiene en diferencias entre cores a través del capacity:
  `capacity(cpu) = work_per_hz(cpu) * max_freq(cpu)`
- Para usuario/as y desarrolladore/as es transparente.
- El scheduler tiene una runqueue por core.
- Periodicamente se chequean desbalances y se redistribuyen tasks.
- Idealmente no se migran tasks a otro core para mantener caches.
- Se puede forzar una asignación de cores para un thread o proceso con cpu affinity:
  - sched_setaffinity()
  - taskset -c

sched_setaffinity()

- Cambia la afinidad de un thread

```
int sched_setaffinity(pid_t pid, size_t
cpusetsize, const cpu_set_t *mask);
```

- pid puede ser el valor devuelto por gettid(), getpid() (aplica al main thread de ese proceso) o 0 (aplica al thread actual)
- cpusetsize: tamaño de la estructura apuntada por mask
- mask: conjunto de CPUs:
  - CPU_SET/CPU_ZERO
  - man CPU_SET

taskset

- Setea la afinidad desde línea de comandos
- taskset -c 0-3 ./comando
- taskset -c 0,4,5 ./comando
- taskset -p <pid>
- taskset -p <mask> <pid>

```
prog@void:~/git/catedra_so/codigo-para-practicas/practica6 [main|…34]
$ taskset -p 1005252
máscara de afinidad actual del pid 1005252: fff
prog@void:~/git/catedra_so/codigo-para-practicas/practica6 [main|…34]
$ taskset -p 0x05 1005252
máscara de afinidad actual del pid 1005252: fff
nueva máscara de afinidad del pid 1005252: 5
```

Fuentes

- https://docs.kernel.org/scheduler/sched-capacity.html
- https://docs.kernel.org/scheduler/sched-design-CFS.html
- https://blog.foool.net/wp-content/uploads/linuxdocs/scheduler.pdf
- manpages : taskset, sched_setaffinity()

¿Preguntas?