



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# SISTEMAS OPERATIVOS

## Práctica 6 - Multiprocesadores

### A - Conceptos teóricos

1. Explique las características principales y formas de comunicación entre procesos en:
  - a. Multiprocesadores con memoria compartida
  - b. Multicomputadora con memoria independiente / pasaje de mensajes
  - c. Sistemas Distribuidos
2. Indique en qué CPU/s se ejecuta el sistema operativo en sistemas de tipo:
  - a. Maestro-esclavo
  - b. SMP
3. ¿Qué puede ocurrir si el kernel S.O. quiere acceder en paralelo a una estructura de datos en la memoria compartida en un sistema SMP? ¿Qué mecanismo permite manejar esta situación?
4. ¿En el caso de una arquitectura SMP puede haber un impacto negativo por migrar un hilo de una CPU a otra?

### B - Práctica guiada

Nota: La práctica requiere ejecutar los programas `affinity` y `affinity_half_and_half` provistos en el repositorio de la cátedra. En cada caso si el programa tarda demasiado o muy poco ajuste la macro `ITERATIONS` y/o la macro `THREADS` para disminuir o aumentar la carga del sistema.

1. Utilice `lscpu` para determinar cuántos cores lógicos tiene su computadora (si usa una máquina virtual asegúrese de configurar al menos 2 CPUs para llevar a cabo el resto de la práctica).
2. Analice el código y comentarios del programa `affinity.c`.
3. Compile los programas provistos en `practica6` con el comando `make`.
4. Ejecute `./affinity` y conteste:
  - a. ¿Qué información muestra el programa?
  - b. ¿Los hilos se ejecutan siempre en el mismo core desde su creación?
  - c. Para más claridad puede elegir un hilo y seguir su ejecución con `grep`:  
`./affinity | grep "thread 4[, ]"`
5. Utilice `taskset` para ejecutar todos los hilos del programa `affinity` en el core 0.
  - a. ¿Cuánto tiempo tardó la ejecución comparativamente con invocar `./affinity` sin `taskset`? Puede usar el comando `time` y sumar los 3 valores que devuelve para obtener un valor preciso.
6. Analice el código y comentarios de `affinity_half_and_half.c`.
  - a. ¿En qué core se ejecutarán los procesos con `rank < THREADS / 2`?
  - b. Ejecute `./affinity_half_and_half` y observe la asignación de cores de forma similar al punto 4. De nuevo puede filtrar un hilo con `grep` para más claridad.

- c. ¿Los hilos que arrancan un core dado siguen toda su ejecución en el mismo core? ¿Por qué?