

# Lab0: precalentamiento

Sistemas Operativos 2014

FaMAF - UNC

Versión original 2014: Nicolás Wolovick, Matías Molina.

## Plazo

1 semana.

## Objetivos Generales

- Comenzar a familiarizarse con el código de `xv6`.
- Empezar a manejar el ciclo edición-compilación-ejecución en `xv6`.
- Programar algo sencillo en *userspace*.
- Utilizar `git` para el ciclo de modificación de `xv6`.
- Proveer de parches de implementación.

## Objetivos Particulares

- Implementar el programa de *userspace* `uptime`.
- Medir la frecuencia de actualización del contador de *uptime*.
- Modificar la programación del PIT o LAPIC para duplicar la frecuencia de la interrupción temporizada y verificar que se aumenta la frecuencia de actualización del contador de *uptime*.

## Primera parte: implementar el comando `uptime`

El sistema operativo [xv6](#) cuenta el tiempo relativo al inicio del sistema, usualmente denominado *uptime*. Para acceder a esta información hay una llamada de sistema (*system call*) denominada del lado del usuario `uptime()` y del lado del kernel `sys_uptime()`.

## Tareas

- Implementar un programa de usuario `uptime` que muestre el valor de la llamada al sistema `uptime()`.
- Deducir y reportar la **frecuencia de actualización** mediante experimentos (cronometrar con un reloj externo).

## Ayudas

- Compilar y ejecutar `xv6`.
- Mirar que programas tiene cargados el sistema de archivos para ser ejecutados en modo usuario.
- Estudiar alguno de los programas *userspace* a fin de inspirarse para escribir el nuevo programa `uptime`.
- También tendrán que modificar el `Makefile`.

## Segunda parte: modificar el temporizador periódico

La interrupción más importante es una interrupción periódica.

Usualmente está generada por el PIT (*programmable interrupt timer*) un [8253](#) para computadoras monoprocesador o por el LAPIC (*local advanced programmable interrupt controller*) para computadoras modernas con multiprocesadores.

Este dispositivo envía una señal al  $\mu P$  cada cierto intervalo de tiempo configurable, y esto produce una interrupción.

Al iniciar el sistema operativo se configura el PIT o LAPIC de manera tal que produzca una interrupción de temporizador típicamente 100 veces por segundo.

### Tareas

- Buscar donde se programa el PIT o APIC según el tipo de arquitectura que estemos utilizando monoprocesador o multiprocesador (SMP - *symmetric multi processing*).
- Cambiar los valores para que produzcan 200 interrupciones por segundo.
- Comprobar el cambio viendo que el comando `uptime` incrementa el doble de rápido.

### Ayudas

- Ver si `xv6` inicializa en modo monoprocesador o multiprocesador. Esto ayuda a definir si la interrupción periódica estará manejada por el PIT o el LAPIC.
- Para asegurarse de que estamos cambiando el temporizador correcto, variar primero en un factor de 10 para asegurarse que funciona decididamente más lento o más rápido.

## Forma de evaluación y entrega

- Este laboratorio es **individual**.
- El **martes 19 de agosto** de 16 a 18 se tomará a cada integrante del grupo como se trabaja sobre `xv6`, `git` y las particularidades de este Laboratorio.
- Deberán tener sobre su repositorio local clonado de [git://pdos.csail.mit.edu/xv6/xv6.git](https://pdos.csail.mit.edu/xv6/xv6.git) los `commit` que implementan el comando `uptime` y la modificación del PIT o LAPIC. El total de los parches no deberían superar las 20 a 30 líneas en total.
- No hay que presentar informe, solo se toma un pequeño examen oral.

### Extras

- ¡Si cambia la base de tiempos la definición de `sys_sleep` cambia! Haga que el tiempo que toma un `sleep(1)` sea independiente de la base de tiempo.
- Pasa exactamente lo mismo con la definición del *quanto*, es decir el período de tiempo que un proceso puede correr sin ser interrumpido. Hacer que el *quanto* y la base de tiempos sean dos configuraciones independientes.

## Manejo básico de `qemu`

- Para listar los procesos dentro de `xv6` hacer `<CTRL-p>`.
- Salir de QEMU: `<CTRL-a> x`.

- Para que inicie la versión correcta de `qemu` escribir antes de `make qemu`:  
`export QEMU="qemu-system-i386 -nographic".`