

## Laboratorio 1 - Sistemas Operativos

### -Integrantes-

Angela Jara	< <a href="mailto:angela.jara@unmsm.edu.pe">angela.jara@unmsm.edu.pe</a> >
Alwin Dávila	< <a href="mailto:alwin.davila@unmsm.edu.pe">alwin.davila@unmsm.edu.pe</a> >
Shamir Mantilla	< <a href="mailto:shamir.mantilla@unmsm.edu.pe">shamir.mantilla@unmsm.edu.pe</a> >

### -Preliminares-

<https://www.scs.stanford.edu/24wi-cs212/pintos/pintos.html>  
[https://grail.eecs.csuohio.edu/~cis345s/PintosCSU\\_Ref.pdf](https://grail.eecs.csuohio.edu/~cis345s/PintosCSU_Ref.pdf)

=====

### ALARM CLOCK

=====

### -DATA STRUCTURES-

A1:

- **struct thread {**  
    **int64\_t wakeup\_tick**  
}
- Almacena el valor del tick en el que un thread debe ser despertado.
- **static struct list sleep\_list**  
Contiene los threads dormidos que esperan ser despertados por **wakeup\_tick**.

### -ALGORITHMS-

A2:

1. Desactiva las interrupciones para evitar race conditions.
2. Calcula el **wakeup\_tick** sumando los ticks actuales con el número de ticks que debe esperar el thread.
3. Inserta el thread en la lista de threads dormidos (**sleep\_list**), ordenada por **wakeup\_tick**.
4. Llama al scheduler para que otro thread se ejecute mientras el thread actual está en estado bloqueado.
5. En cada tick, el timer interrupt handler llama a **wakeup\_threads()**, que revisa la **sleep\_list** y despierta los threads cuyo **wakeup\_tick** ha llegado.

A3:

- Se realizan solo las operaciones esenciales, como incrementar el contador de ticks y verificar si un thread debe despertarse.
- **Sleep\_list** se mantiene ordenada por **wakeup\_tick**, permitiendo una verificación rápida del thread que debe despertar.
- Los threads se despiertan en grupos si tienen el mismo **wakeup\_tick**.

## **~~—SYNCHRONIZATION—~~**

**A4:**

Las interrupciones se desactivan mientras se inserta el thread en **sleep\_list**. Esto asegura que ningún otro thread ni el interrupt handler modifiquen la lista al mismo tiempo.

**A5:**

Se deshabilitan las interrupciones al inicio de **timer\_sleep**. Esto previene que el interrupt handler interfiera en lo que se pone a dormir un thread.

## **===== PRIORITY SCHEDULING =====**