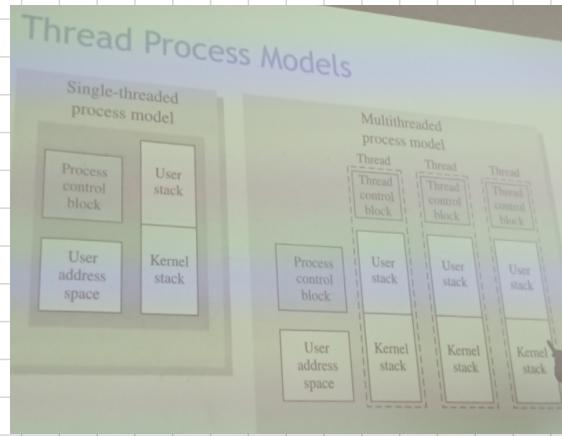
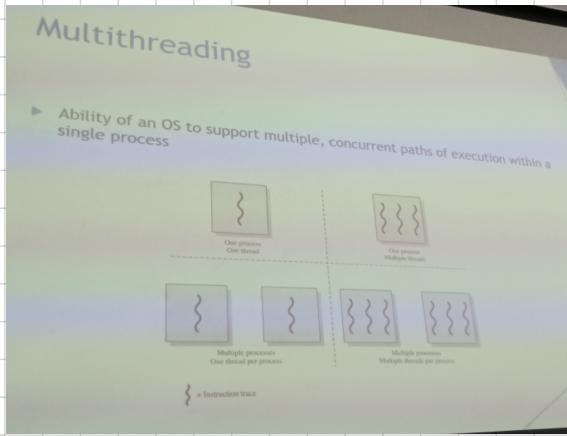


- Resource ownership
- Scheduling Execution
- Independent characteristics
- Unit of resource ownership:
  - ▶ Process
  - ▶ Task
- Unit of scheduling:
  - ▶ Thread
  - ▶ Light-weight



- Threads benefits:
    - Creation time
    - Termination time
    - switching time
    - Enhanced efficiency in communication mscambiar variables del proceso
    - Simplify the structure of a program that is logically doing several different functions
  - Threads states:
    - Rotar procesador entre threads
    - Funcionan en 5 estados
    - Process states vs Thread states
      - Ready: All threads in ready or blocked } process is almost running
      - Running: Any thread in running at all times.
      - Blocked: All threads in blocked
    - Fair execution time:
      - Quantum time variable mscambiar si procesos tienen más threads, mayor uso de procesador
        - ↳ Menos threads mayor quantum
        - ↳ Mas threads menor quantum
  - Threads types:
    - Kernel level → Los crean al so mscambiar diferentes a cada plataforma!  $\rightarrow WN \neq Linux \neq Mac OS$
    - User level threads → Se usa una biblioteca y constan threads pareados.
      - {} L → E.g. JDK - Java mscambiar para cada os
- Intercambio lo hace la biblioteca  $\rightarrow$  Generamos interacciones



Suponiendo los siguientes eventos en el tiempo (en milisegundos):

- Creación del proceso A y su Thread principal (T=0).
- Creación del thread T1 en proceso A (T=100).
- Proceso A (Thread T1) solicita uso del periférico 1 (T=110).
- Creación del proceso B y su Thread principal (T=200).
- Proceso B (Thread principal) solicita uso del periférico 2 (T=205).
- periférico 1 termina operación (T=1000).
- Proceso A (Thread principal) solicita uso del periférico 2 (T=1005).
- Proceso B (Thread principal) solicita uso del periférico 1 (T=3015).
- periférico 2 termina operación (T=3000).
- periférico 1 termina operación (T=4200).
- periférico 2 termina operación (T=5000).

Si un proceso se suspende si lleva un milisegundo bloqueado, el "scheduler" retoma el control con una interrupción de reloj cada 0.1 milisegundos y cada proceso requiere 12000 milisegundos de procesador en su thread principal. Un proceso también se suspende si ha tenido el procesador por 3000 ms o más (desde que inicio ejecución o desde la última suspensión) y permanece suspendido 200 ms.)

## 1. Usando procesos de 5 estados , especifique para cada proceso:

- Tiempo total en el sistema operativo.
- Tiempo de ejecución en el procesador (estado ejecutando).
- Tiempo en estado listo.
- Tiempo en estado bloqueado.
- Tiempo en estado suspendido.
- Tiempo en estado listo-suspendido.
- Tiempo en la cola de eventos esperando por un periférico.
- Tiempo que demora cada periférico en completar cada petición.

• Igual que el anterior, sobre los threads