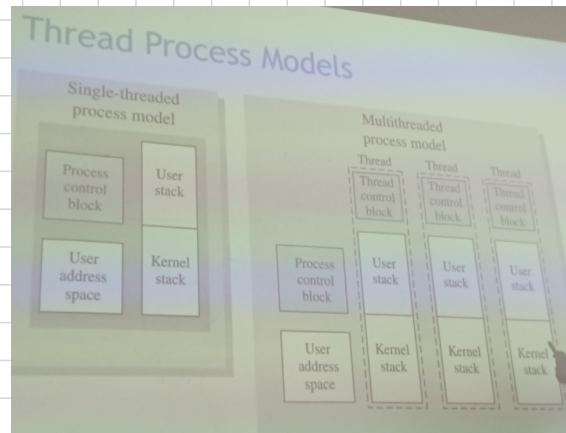
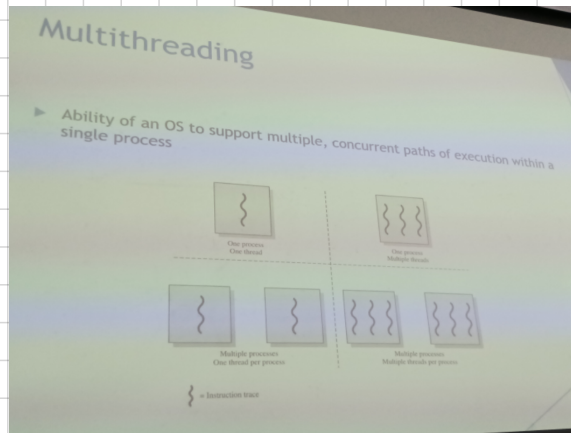


- Resource ownership
- Scheduling Execution
- Independent characteristics
- Unit of resource ownership:
 - Process
 - Task
- Unit of Scheduling:
 - Thread
 - Lightweight



- Threads benefits:
 - Creation time
 - Termination time
 - Switching time
 - Enhanced efficiency in communication *ms Cambiar variables del proceso*
 - simplify the structure of a program that is logically doing several different functions

• Threads Status:

- Rotar procesador entre threads
- Funcionan en 5 estados
- Process status vs Thread Status
 - 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 *seida* *ms* Si procesos tienen más threads, mayor uso de procesador

↳ Menos threads mayor quantum
↳ Mas threads menor quantum

• Threads types:

- Kernel level *→ Los crea el so ms Diferentes a cada plataforma! → WN ≠ Linux ≠ Mac OS*
 - User level threads *→ Se usa una biblioteca y crean threads parecidos.*
 - ↳ Ej: JOK - Java ms Diferente para cada OS*
- Intercambio lo hace la biblioteca *ms* Generamos inofinencias



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 el thread principal. Un proceso también se suspende si ha tenido el procesador por 3000 ms o más (desde que inició ejecución o desde la última suspensión) y permanece suspendido 200 ms.)

• Igual que el anterior, sobre los threads

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.