

# ***Sistemas Operativos***

***Cursada 2022***

**Comisión S21 y S22**

# *Sincronizacion*

Temas tratados:

- **Multiprocesamiento**
- **Multiprogramación**
- **Cooperativos**

Problema que no existe en los sistemas mono proceso.

***Es la Sincronización de Procesos***

# ***Sincronización***

- ***Procesos comparten recursos***
- ***Pueden ser adquiridos por mas de uno***
- ***Procesos en ejecucion concurrente***
- ***Procesos que corren en multi CPU***

# *Sincronización*

Supongamos que hay un recurso compartido en un sistema determinado.

Ese recurso supongamos que es una variable, como ejemplo puede ser “**stock-PC**”

**El resultado de una operación depende del orden en que se ejecute, esto se llama**

## **Condición de Carrera**

**Caso muy serio en programación concurrente**

# ***Sincronización***

**Dentro de la codificación denominamos**

## ***Sección Crítica***

**A la parte del código del programa (en nuestro caso “stock-pc” ) en la cual se va acceder y modificar un recurso compartido**

## **Para evitar problemas**

**Si un proceso esta en la sección crítica otro no puede estar ejecutando esa sección crítica que afecta a ese recurso.**

# *Sincronización*

Procedure P1;

Begin

While condición 1 do

Begin

**Actividades Preliminares\_1**

**Sección Crítica;**

**Actividades Posteriores\_1**

End

End

# *Sincronización*

Tener en cuenta que puede haber varias secciones críticas en un programa.

```
If stock_PC > 0
{
    actividades preliminares
    stock_pc -= 1
    actividades posteriores
}
```

**stock\_pc -= 1** Asi se escribe en un lenguaje de alto nivel

Que pasa a bajo nivel??

# ***Sincronización***

- **Existen distintas soluciones a este problema**
- **Ahora que condiciones debe tener o cumplir la solución que se proponga al problema de la sección crítica??**
- **Debemos definir un Protocolo**
- **Debemos saber si estamos en sistemas mono CPU o multiple CPU**
- **Plantearemos un ejemplo para entenderlo**



# *Sincronización*

## ➤ Exclucion Mutua – **Mutex** -(uno solo en la S.C.)

- Necesitamos un protocolo
- Cerrojo
- Funciona para una cola FIFO
- Planteo de (Stallings – Tanenbaum)
- Ahora funciona para todos los procesos??
- Que pasa si el recurso se asigna por prioridad
- Planteo que se hace “Silberschatz”
- Quienes participan en la decisión
- Dado que puede ocurrir un caso de “**inanición**”

# ***Sincronización***

- **Progreso** (Silberschatz) Solo participan los procesos que están en la cola (no lo que esta en la sección restante)
- **Espera limitada** (Silberschatz) Ningun proceso puede esperar ilimitadamente la entrada a la seccion critica

# ***Sincronización***

Existen varias soluciones a este problema

## ➤ **Soluciones por Software**

- Algoritmo de Peterson
- Se puede implementar en todo sistema?
- Que pasa en los sistemas multi CPU

## ➤ **Soluciones por Hardware**

Se implementaron instrucciones a nivel del procesador

- Test-and-set (**verificar y colocar el valor**)
- Swap
- Realizan en un solo ciclo de instrucción toda la operación
- No se pueden interrumpir

# *Sincronización*

➤ Herramientas específicas para esto  
implementadas a nivel Sistema Operativo

- **Semaforos**

- **Monitores**

***Fin del Tema***