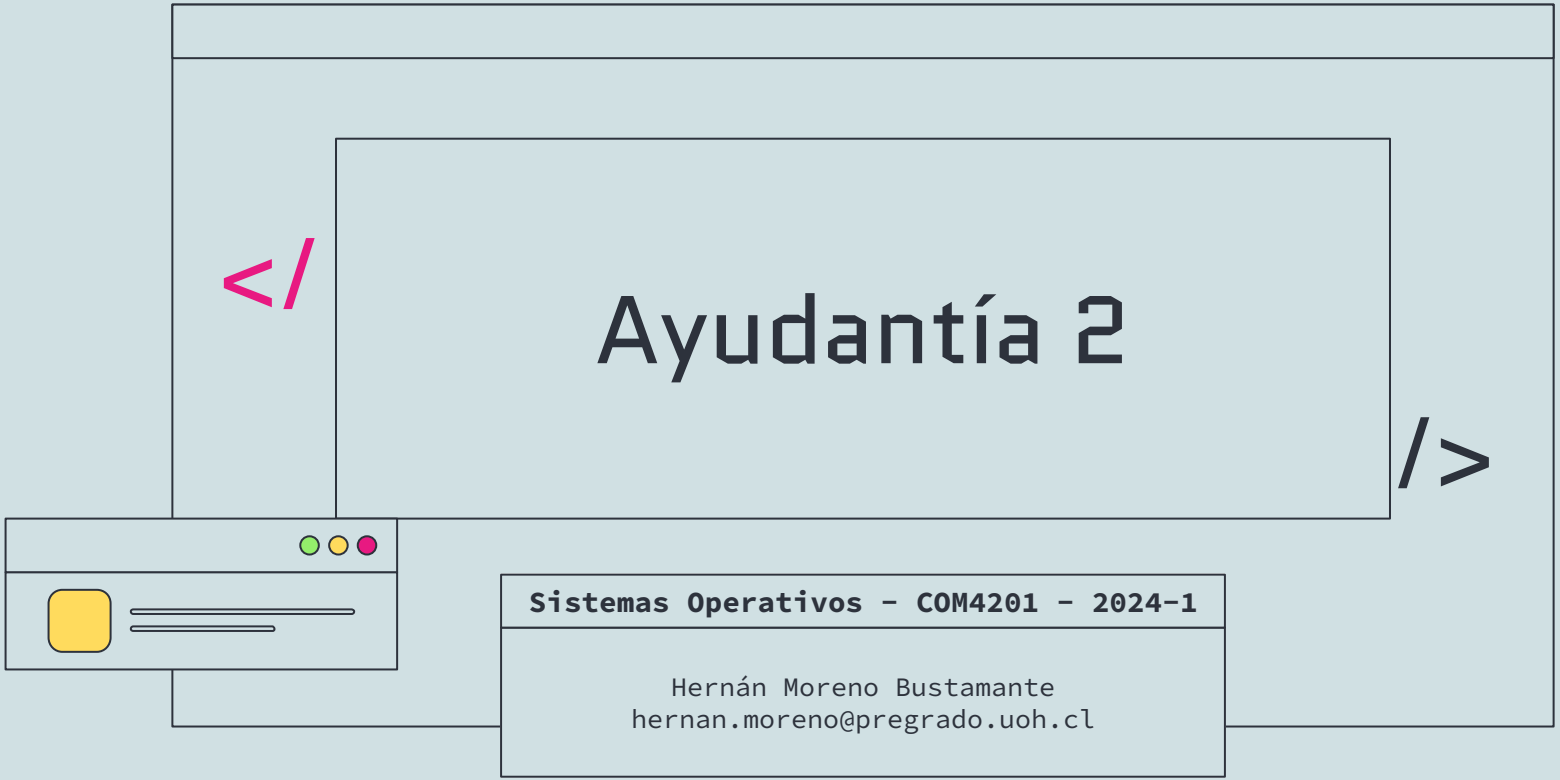


Sistemas Operativos - COM4201 - 2024-1

Hernán Moreno Bustamante
hernan.moreno@pregrado.uoh.cl

Recordatorio

</Semaforos

nMakeSem(int n)	Crea el semáforo con n tickets iniciales
nWaitSem()	Pide un ticket
nSignalSem()	Entrega un ticket al semáforo, si hay procesos esperando se atienden en orden FIFO

</Monitores

nMakeMonitor( )	Construye el monitor
nEnter(nMonitor m)	Indica el inicio de la sección crítica
nExit(nMonitor m)	Indica el final de la sección crítica
nWait(nMonitor m)	Libera el monitor m y suspende la tarea que lo invoca hasta que otra tarea invoque nNotifyAll
nNotifyAll(nMonitor m)	Notifica a todos (los que están esperando) que se liberó el monitor m
nDestroyMonitor(nMonitor m)	Destruye el monitor m

</Problema 0 (opcional)

- Utilizando la herramienta de sincronización semáforos, sincronice el proceso escritor y con los lectores.

</Monitores de Hoare

nMakeCondition(nMonitor mon)	Construye la condición, asignandola al monitor mon
nDestroyCondition(nCondition cond)	Destruye la condición cond
nWaitCondition(nCondition cond)	Suspende una tarea como lo hace el nWait
nSignalCondition(nCondition cond)	Despierta a las tareas con la condición cond.

</Problema 1

- Suponga que posee un negocio pequeño atendido por **N** vendedores. Al llegar, los clientes obtienen un número de atención y esperan a que el vendedor los llame a atenderlo. Cuando un vendedor se desocupa, llama al siguiente número. Simule los procesos de los **N** vendedores y los **M** clientes utilizando monitores estilo Hoare.

</Mensajes

nSend(nTask pid, *msg)	Enviar un mensaje (msg) a otro proceso destinatario (pid).
nReceive(nTask *pid, int t)	Espera a que llegue un mensaje y guarda en el pid local, el pid del remitente. Espera t segundos, usar número negativo para que espere indefinidamente.
nReply(nTask pid, int rc)	Responde al mensaje recibido, confirmación de haberlo recibido. Rc es el código de retorno.

</~~Problema 2~~ Ejemplo de mensaje

- Ejemplo de productor/consumidor sincronizados mediante mensajes.

- **Propuesto:** aumentar tamaño del buffer

</Resumen

Semaforos	**Monitores**

`nSem sem; // = nMakeSem(n);`
 	
`int fun(){`
`  ...;`
`  nWaitSem(sem);`
`  // sección crítica`
`  nSignalSem(sem);`
`  ...;`
`}`
	`nMonitor m; // = nMakeMonitor();`
	
	`int fun(){`
	`  ...;`
	`  nEnter(m);`
	`  while(condicion){`
	`  nWait(m);`
	`  }`
	`  // sección crítica`
	`  nNotifyAll();`
	`  nExit(m);`
	`  ...;`
	`}`

</Resumen

Monitores Hoare

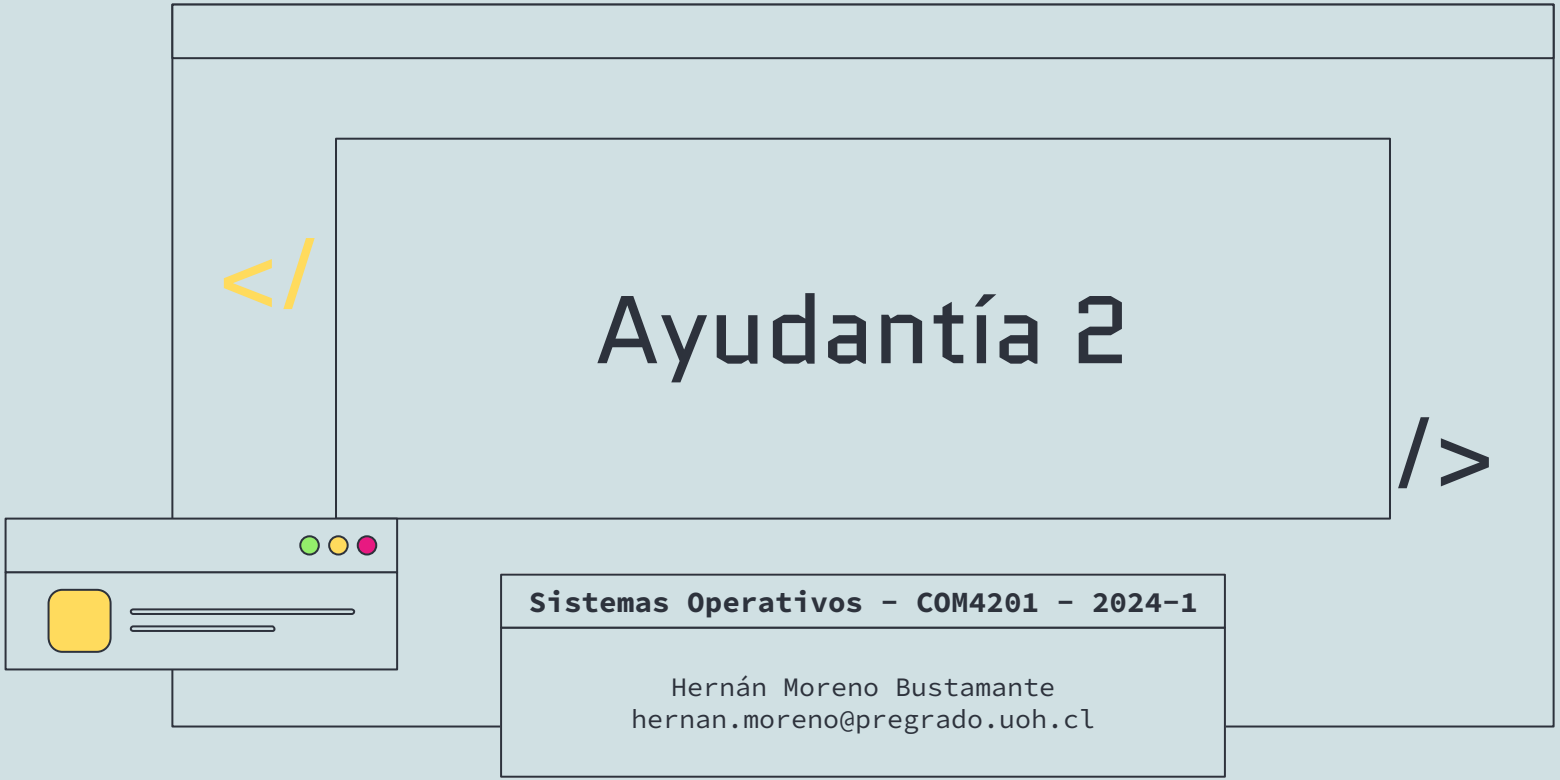
```
nMonitro m;
nCondition = cond1, cond2;

int fun(){
  ...;
  nEnter(m);
  while(condicion){
    nWaitCondition(cond1);
  }
  // sección crítica
  nSignalCondition(cond2);
  nExit(m);
  ...;
}
```

Mensajes

```
int fun1(){
  Item it;
  ...;
  nSend(dest, &it);
  ...;
}
```

```
int fun2(){
  nTask t;
  Item *p_it= (Item*)nReceive(&t, -1);
  ...;
  nReply(dest, rc);
  ...;
}
```


Ayudantía 2
Sistemas Operativos - COM4201 - 2024-1
Hernán Moreno Bustamante
hernan.moreno@pregrado.uoh.cl