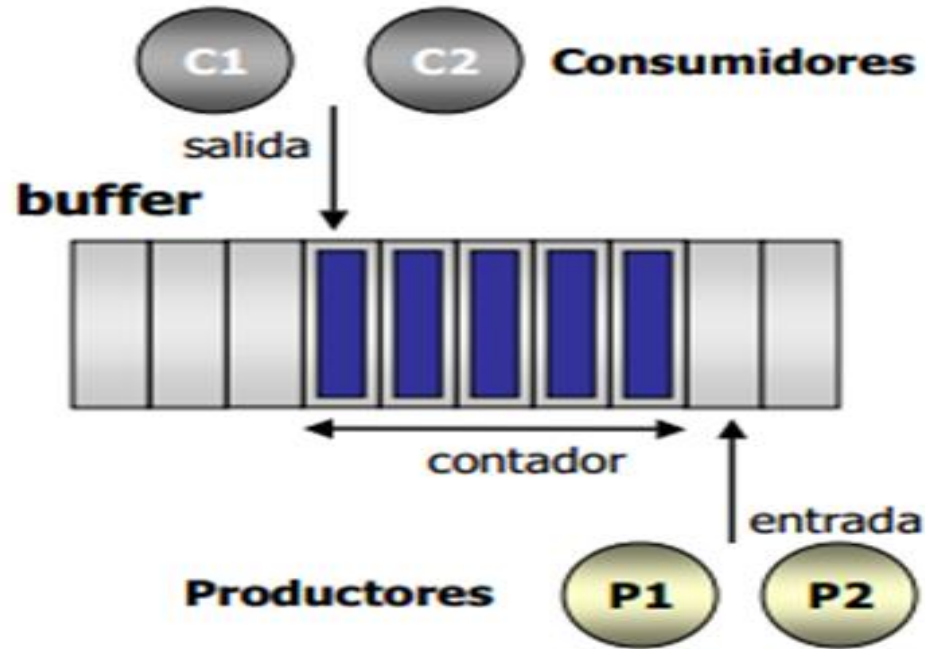


PROGRAMACIÓN CONCURRENTE CON SEMÁFOROS Y MONITORES

Sistemas Operativos

Prof. PhD Mirella Herrera

Productor-Consumidor con Semáforos



Productor-Consumidor con Semáforos

```
int contador = 0; //indica número de items en buffer
char buffer[N];
int in = 0; int out = 0;
sem mutex=1; sem vacio = N; sem lleno = 0;
```

Productor

```
while (true) {
    /* produce un item en proxProd */
    wait(vacio);
    wait(mutex);
    buffer[in] = proxProd;
    in = (in + 1) % N;
    contador++;
    signal(mutex);
    signal(lleno);
}
```

Consumidor

```
While(true){
    wait(lleno);
    wait(mutex);
    proxCons = buffer[out];
    out = (out + 1) % N;
    contador--;
    signal(mutex);
    signal(vacio);
    /* consume proxCons */
}
```

Productor-Consumidor con Monitores

monitor ProdCons

```
condition lleno, vacio;  
integer cont, N;
```

```
procedure producir;  
begin  
    if (cont = N) then  
        wait(lleno);  
    introducir_elemento;  
    cont := cont + 1;  
    if (cont = 1) then  
        signal(vacio);  
end;
```

```
procedure retirar;  
begin  
    if (cont = 0) then  
        wait(vacio)  
    retirar_elemento;  
    cont := cont - 1;  
    if (cont = N - 1) then  
        signal(lleno);  
end;
```

```
cont := 0; N = 100;  
end monitor;
```

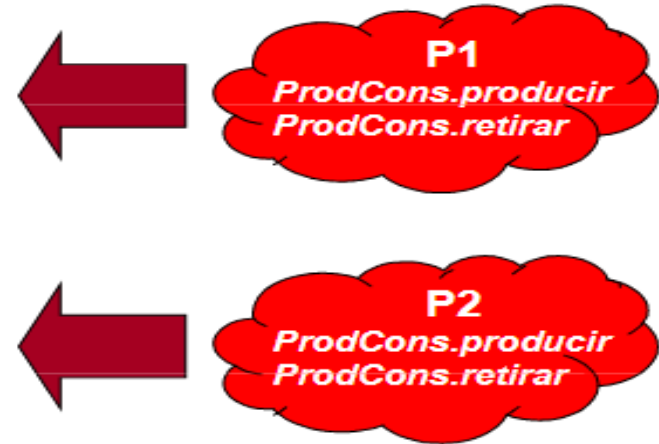
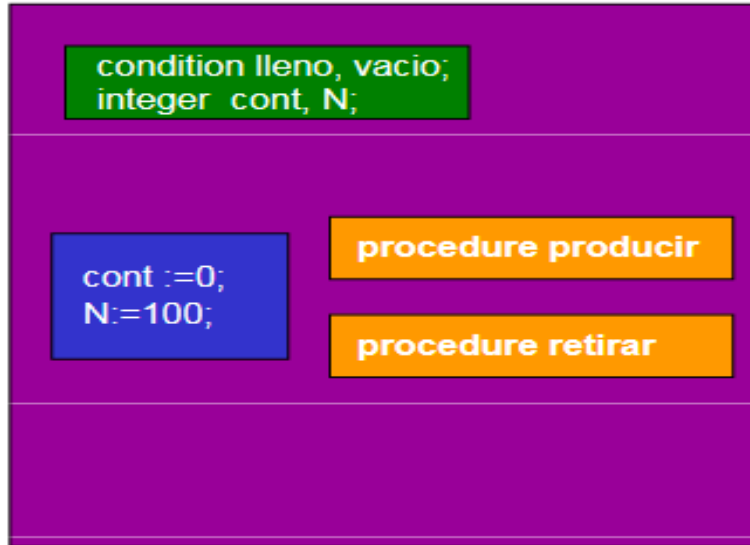
Producer-Consumidor con Monitores

```
procedure productor;  
begin  
  while true do  
  begin  
    producir_elemento;  
    ProdCons.producir;  
  end  
end;
```

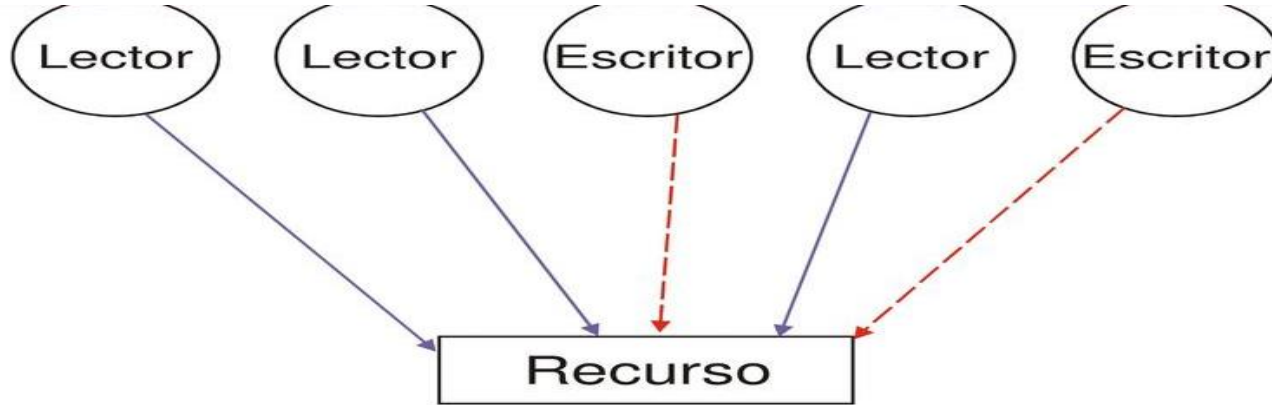
```
procedure consumidor;  
begin  
  while true do  
  begin  
    ProdCons.retirar;  
    consumir_elemento;  
  end  
end;
```

Productor-Consumidor con Monitores

monitor ProdCons



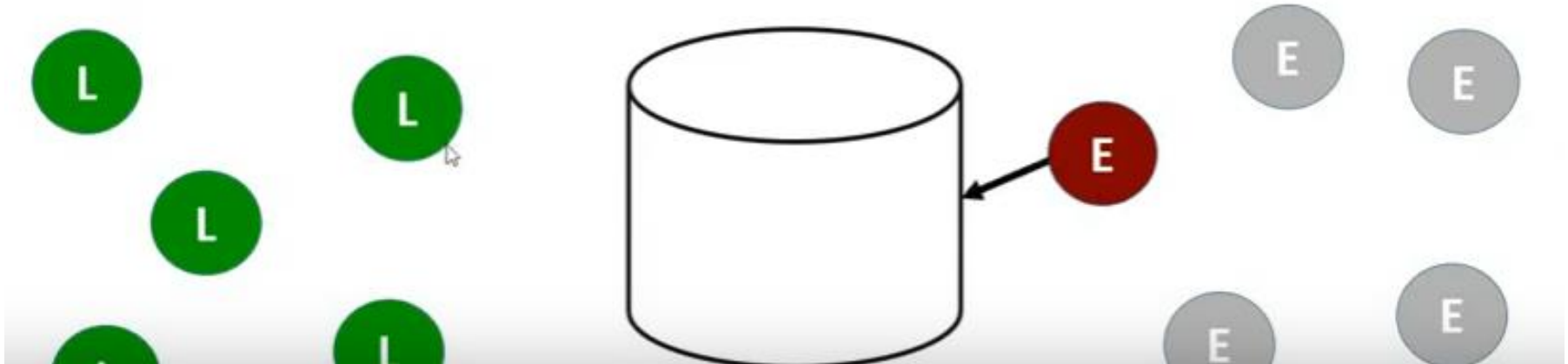
Lectores-Escritores



- Se da una combinación de procesos que leen y escriben sobre el recurso. Se dan las siguientes restricciones:
 - Sólo se permite acceso a un escritor al mismo tiempo. Ningún proceso lector o escritor puede acceder.
 - Se permite, que múltiples lectores tengan acceso al objeto ya que estos no modifican el contenido.

Lectores-Escritores con prioridad de los Escritores

No permitir acceder a los datos a ningún nuevo lector una vez que, un escritor haya declarado su deseo de escribir.



Lectores-Escritores con prioridad de los Escritores (semáforos)

```
int contlect=0,contesc=0;
Semaphore x,y,z,esem,lsem;
Lector()
{
    while(forever)
    {
        wait(z);
        wait(lsem);
        wait(x);
        contlect++;
        if(contlect==1) wait(esem);
        signal(x);
        signal(lsem);
        signal(z);
        LEER_UNIDAD;
        wait(x);
        contlect--;
        if(contlect==0) signal(esem);
        signal(x);
    }
}
```

```
Escritor()
{
    while(forever)
    {
        wait(y);
        contesc++;
        if(contesc==1) wait(lsem);
        signal(y);
        wait(esem);
        ESCRIBIR_UNIDAD;
        signal(esem);
        wait(y);
        contesc--;
        if(contesc==0) signal(lsem);
        signal(y);
    }
}
```

```
main()
{
    initsem(x,1);initsem(y,1);initsem(z,1);
    initsem(esem,1);initsem(lsem,1);
    cobegin {
        Lector(); Escritor();
    }
}
```