## *SEMAFORO BINARIO*

```
void P (struct x)
      if x.value = 1 //se libera
            x.value = 0
      sospendi(me, x.coda) //accodo
rilascia
```

P

```
void V (struct x)
      x.value = 1 //rilascio
      if not_empty(x.coda)
            ready(get(x.codaI)
rilascia
```

V

***ES**: Lavoro sulla stessa variabile*

***PROC. A***
*<<esegui comando indipendente>>*
*P(x) //richiedo risorsa*
*<<eseguo comando in lock>>*
*V(x) //rilasico*

***PROC. B***
*P(x) //richiedo risorsa*
*<< se occupata aspetto >>*
*<< inizio >>*
*V(x) //rilascio dopo l'uso*

## *SEMAFORO CONTATORE*

```
void P (struct x)
      x.value -- //tolgo unità di risorsa
      if x.value < 0 //se unità insuff
      sospendi(me, x.coda) //accodo
rilascia
```

P

```
void V (struct x)
      x.value ++
      if x.value <=0  //se unità insuff
            ready(get(x.codaI)
rilascia
```

V

***Es**: produttore e compratore*

```
#define N 100
Typedef int x
x mutex = 1
x n-v = 0
x n-p = N
```

***PRODUTTORE***
```
Int produci
      While (1)
        Produci:=prod()
        P(&n-p)          //fabbrica lock-on
        P(&mutex)        //lock-on
        Inserisci()
        V(&mutex)        //lock-off
        V(&n-v)          //magazzino lock-off
```

***CONSUMATORE***
```
Int prod
      While (1)
        P(&n-v)          //magazzino lock-on
        P(&mutex)        //lock-on
        Preleva()
        V(&mutex)        //lock-off
        V(&v-p)          //fabbrica lock-on
        Consuma()
```

## *MONITOR*

***PRODUTTORE***
```
procedure Prod
      begin
      while true do begin
            prod:=produci
            PC.inserisci(prod)
      end
```

***CONSUMATORE***
```
procedure Prod
      begin
      while true do begin
            prod:=PC.preleva
            consuma(prod)
      end
```

***MONITOR***
```
Monitor
      condition: n-v, n-p
      int cont = 0

procedure inserisci
      if cont = N
         wait(n-p)
      <<inserisci>>
      cont ++
      if cont = 1
         signal(n-v)

procedure preleva
      if cont = 0
         wait(n-v)
      preleva:= <<preleva>>
      cont –
      if cont = N-1
         signal(n-p)
      end
```