

Semaphores - definition

- Proposed by Dijkstra, it was the first high level constructs used to synchronize concurrent processes.
- A semaphore S is an integer variable on which two atomic operations are defined, $P(S)$ and $V(S)$, and with an associated queue.
- P and V semantic:

```
P(S): if  $S \geq 1$  then  $S := S - 1$   
      else <block and enqueue the process>;
```

```
V(S): if <some process is blocked on the queue> then  
      <unblock a process>  
      else  $S := S + 1$ ;
```

Semaphores - properties

- The P operation may block a process, but V does not
- Two type of semaphores
 - binary: intial value is 1
 - resource counting: any initial value
- P and V are atomic operations

```
P(S): if  $S \geq 1$  then  $S := S - 1$   
      else <block and enqueue the process>;
```

```
V(S): if <some process is blocked on the queue> then  
      <unblock a process>  
      else  $S := S + 1$ ;
```

Example of use

Shared var mutex: semaphore = 1;

Process *i*

```
begin
.
.
P(mutex);
execute CS;
V(mutex);
.
.
End;
```

Other synchronization problems

- Semaphore can be used in other synchronization problems besides Mutual Exclusion
- The Producer-Consumer problem
 - a finite buffer pool is used to exchange messages between producer and consumer processes
- The Readers-Writers Problem
 - reader and writer processes accessing the same file
- The Dining Philosophers Problem
 - five philosophers competing for a pair of forks



Reader-Writers problem

- The shared resource is a file accessed by both reader and writer processes
- The synchronization constraints are:
 - readers should be able to concurrently access the file
 - only one writer at a time can access the file
 - readers and writers exclude each others
- Variants:
 - reader's priority: arriving readers have priority over waiting writers
 - writer's priority: writers have priority over waiting readers

ERROR: invalidaccess
OFFENDING COMMAND: def

STACK:

```
3  
/XComps  
false  
(  
  yyyyyyyyyyyyyyyyyyyyyyyyyyy  yyyyyyyyyyyyyyyyyyyyyyy  yyyyyyyyyyyyyyyyyyyyyyy  yyyyyyyyyyyyyyyyyyy  
)  
[1 0 0 1 0 0 ]  
8  
64  
64
```