

## 1994 Paper 5 Question 7

### Concurrent Systems

Semaphores are used to achieve exclusive access to a resource for writing but to allow simultaneous read-only access. *Reader* and *Writer* processes have the following structure:

```
call a procedure to acquire the resource;
use the resource;
call a procedure to release the resource.
```

The procedures to acquire the resource *for reading* and to release it are as shown below. Writers have priority over readers.

```
procedure acquire_for_reading()
  WAIT(CGUARD);
  ar := ar + 1;      # increment the count of active readers
  if aw = 0 then    # if there are no active writers
    begin
      rr := rr + 1; # increment the count of reading readers
      SIGNAL(READ)
    end;
  SIGNAL(CGUARD);
  WAIT(READ);
```

```
procedure release_after_reading()
  WAIT(CGUARD);
  rr := rr - 1;
  ar := ar - 1;
  if rr = 0 then
    while ww < aw do # ww counts "writing writers"
      begin
        ww := ww + 1;
        SIGNAL(WRITE)
      end;
    SIGNAL(CGUARD);
```

```
# note that "writing writers" are assumed to take turns to use
# the resource.
```

## 1994 Paper 5 Question 7 (continued)

What is the function of the operations on the semaphore CGUARD? [3 marks]

Why is the following, apparently simpler, version of `acquire_for_reading()` incorrect?

```
WAIT(CGUARD);  
ar := ar + 1;  
if aw  $\neq$  0 then WAIT(READ);  
rr := rr + 1;  
SIGNAL(CGUARD);
```

 [3 marks]

What is the function of the operations on the semaphores READ and WRITE? [4 marks]

The procedures to acquire the resource for writing and to release it after writing are similar to those given but, in addition, exclusive access to the resource must be enforced. Outline how this could be programmed. [2 marks]

Write the procedure `release_after_writing()`. [8 marks]