Solutions for CP Exercises, October 30

1. Solution for Andrews Ex. 8.9

(a) A solution giving priority to writers is shown in [Andrews p.388] using the facility to query the number pending calls of an operation *op* to block readers when there are pending writers.

(b) A fair solution can be obtained from the above solution by explicitly processing the readers inbetween the writers. Using our notation, we get:

```
module ReadersWriters
  op read(var T);
  op write(T);
body

  op startread();
  op endread();

  var val : T;

  proc read(var r : T)
    startread();
    r := val;
    endread()

  process Writer =
    var nr : integer := 0;
    repeat
      in startread() and ?write = 0 → nr := nr + 1
      [] endread()                  → nr := nr − 1
      [] write(v : T) and nr = 0    → val := v
                                      while ?startread > 0 do
                                        in startread() → nr := nr + 1 ni

      ni
    forever;

end ReadersWriters;
```

2. Solution for Andrews Ex. 8.10

If

call `FileServer[i].remote_write`(values)

is replaced by

send `FileServer[i].remote_write`(values)

the solution will no longer work. Since `remote_write` is serviced by RPC, the remote updates on the same node may now overlap. Even if the updates were serviced in order by a write server, the write lock might still be released before all updates were done and subsequent read would not see the proper values.