

Copyright Notice

The following manuscript

EWD 504: Erratum and embellishments of EWD503

is held in copyright by Springer-Verlag New York.

The manuscript was published as pages 145–146 of

Edsger W. Dijkstra, *Selected Writings on Computing: A Personal Perspective*,
Springer-Verlag, 1982. ISBN 0-387-90652-5.

**Reproduced with permission from Springer-Verlag New York.
Any further reproduction is strictly prohibited.**

Erratum and embellishments of EWD503.

Erratum: the text of the procedure "release" on page EWD503 - 2 should begin as follows.

```
proc release:
  if busy → busy:= false;
    upsweep:= (upsweep, qu1); downsweep:= (downsweep, qu2);
    if ...etc.
```

To keep the interpunction consistent, I should have used a colon in line 6:

```
proc request(dest: cylinder):
      *           *
      *           *
```

First embellishment: the text of the procedure "endwrite" on page EWD503 - 1 is no longer "quite nice", since I discovered the alternative:

```
proc endwrite:
  if aw = 1 → aw:= 0; writers:= (head(readers), writers);
    me:= head(writers)
  fi
corp endwrite;
```

When there are both readers and writers waiting, it avoids the final unnecessary activation of the oldest writer. Clearly, "shunting" is something I still have to learn!

```
*           *
*           *
```

Second embellishment: C.S.Scholten pointed out to me, that the dishead monitor of C.A.R.Hoare, and therefore also the one on page EWD503 - 2 has on a macroscopic scale a danger of individual starvation. If direction = up and the train "upsweep" is not empty --more precisely: contains requests with dest > headpos-- a continuous stream of requests with dest = headpos can cause the requests in upsweep never to honoured. The moral of the story is that requests with dest = headpos have to be placed in the other stream! The remedy seems to be, to replace line 9 by

```
if dest > headpos or dest = headpos and direction = down →
```

and line 17 by

```
|| dest < headpos or dest = headpos and direction = up →
```

Wasn't that a nice pitfall? And then to think that there are still people that still refuse to believe that programming is difficult.....

```
*           *
*           *
```

Remark about the devious influence of the programming language we are using: if I had been trained to think in PL/I with its horrible "BEGIN statement" "END statements" and "RETURN statements" the invention as described in EWD501 in which the notion of "the dynamically last statement of a monitor procedure" plays a role, would probably not have been made! Again a frightening thought!