4 Remote invocation and selective waiting

Just as when for synchronous message-passing, when remote invocation is used it is helpful to have a way to choose between alternatives:

```
process share;
    entry read(var i : integer);
    entry write(i : integer);

    var
    value : integer;

    begin
    accept write(i : integer) do
        value := i;

    repeat
        select
            accept write(i : integer) do
                value := i;
        or
            accept read(var i : integer) do
                i := value;
        end;
    forever;
    end;
```

Note that multiple `accept`s for the same entry are allowed. In the example above this feature has been used to ensure that the value of the shared integer is set before it can be accessed.

Note also that the read call is made by the client to the server, yet the information flow during this call is from the server to the client.

Just as for the synchronous message-passing `select` statement, a remote invocation `select` statement can have

- else
- timeout
- terminate
- guarded alternatives

In general, the alternatives in a remote invocation `select` statement could include entry calls as well as `accept`s, though Pascal-FC does not allow entry calls in `select` statements. But as we've seen, data can flow in either direction during an entry call so this is not a problem.