

7.3. An Example: Unbuffered Producer-Consumer

Because the send and receive operations effect a rendezvous, simple process synchronisation is straightforward. In the following program, a producer generates characters which are then consumed by a second process.

```

program procon1;

var link: channel of char;

process producer;

var local: char;

begin
  repeat
    (* generate character *)
    link ! local
  forever
end; (* producer *)

process consumer;

var local: char;

begin
  repeat
    link ? local;
    (* consume character *)
  forever
end; (* consumer *)

begin (* main *)
  cobegin
    producer;
    consumer
  coend
end.

```

7.4. Process States and Transitions

This section summarises the effects on process states of the features described in this chapter.

1. A process that attempts a send or receive operation on a channel on which there is no pending call becomes "suspended" if the channel is not mapped to a source of