- Σχήμα 5.2a

P0	**P1**
while (turn != 0) /**/;	while (turn != 1) /**/;
critical section	*critical section*
turn = 1;	turn = 0;

  - Shared variable *turn* indicates who is allowed to enter next, can enter if *turn* = *me*
  - On exit, point variable to other process
  - Deadlock if other process never enters

- Σχήμα 5.2b

P0	**P1**
while (flag[1]) /**/;	while (flag[0]) /**/;
flag[0] = true;	flag[1] = true;
critical section	*critical section*
flag[0] = false;	flag[1] = false;

- Σχήμα 5.2c

P0	**P1**
flag[0] = true;	flag[1] = true;
while (flag[1]) /**/;	while (flag[0]) /**/;
critical section	*critical section*
flag[0] = false;	flag[1] = false;

- Σχήμα 5.2d

P0	**P1**
flag[0] = true;	flag[1] = true;
while (flag[1])	while (flag[0])
{	{
flag[0] = false;	flag[1] = false;
delay	*delay*
flag[0] = false;	flag[1] = false;
}	}
critical section	*critical section*
flag[0] = false;	flag[1] = false;

- Αλγόριθμος Peterson (Σχήμα 5.3b)

P0	**P1**
flag[0] = true;	flag[1] = true;
turn = 1;	turn = 0;
while (flag[1] && turn == 1) /**/;	while (flag[0] && turn == 0) /**/;
critical section	*critical section*
flag[0] = false;	flag[1] = false;

- Παραγωγός (Σχήμα 5.13)

```
do forever
    produce item
    wait(s)
    append to queue
    n++
    if n = 1 then signal(delay)
    signal(s)
```

- Καταναλωτής

```
wait(delay)
do forever
    wait(s)
    remove from queue
    n--
    m = n
    signal(s)
    if m = 0 then wait(delay)
```

Σχήμα 5.20

```
program     barbershop2;
var         max_capacity: semaphore (:=20);
            sofa: semaphore (:=4);
            barber_chair, coord: semaphore (:=3);
            mutex1, mutex2: semaphore (:=1);
            cust_ready, leave_b_chair, payment, receipt: semaphore (:=0)
            finished: array [1..50] of semaphore (:=0);
            count: integer;

procedure customer;
var custnr: integer;
begin
   wait (max_capacity );
   enter shop;
   wait( mutex1 );
   count := count + 1;
   custnr := count;
   signal( mutex1 );
   wait( sofa );
   sit on sofa;
   wait( barber_chair );
   get up from sofa;
   signal( sofa );
   sit in barber chair;
   wait( mutex2 );
   enqueue1( custnr );
   signal( cust_ready );
   signal( mutex2 );
   wait( finished[custnr] );
   leave barber chair;
   signal( leave_b_chair );
   pay;
   signal( payment );
   wait( receipt );
   exit shop;
   signal( max_capacity );
end;

procedure barber;
var b_cust: integer
begin
   repeat
      wait( cust_ready );
      wait( mutex2 );
      dequeue1( b_cust );
      signal( mutex2 );
      wait( coord );
      cut hair;
      signal( coord );
      signal( finsihed[b_cust] );
      wait( leave_b_chair );
      signal( barber_chair );
   forever
end;

procedure cashier;
begin
   repeat
      wait( payment );
      wait( coord );
      accept payment;
      signal( coord );
      signal( receipt );
   forever
end;
```

Dr. Garmpis Aristogiannis
- EPDO TEI Messolonghi

Σχήμα 5.28

- Semaphores: *x*, *wsem* = 1
- Reader

```
wait(x);
readcount++;
if  ( readcount == 1 ) wait(wsem);
signal(x);
Reader C.S.
wait(x);
readcount--;
if ( readcount==0 ) signal(wsem);
signal(x);
```

- Writer

```
wait(wsem);
Writer C.S.
signal(swim);
```

Σχήμα 5.29

- Reader

```
wait(z);   wait(rsem);   wait(x);
readcount++;
if ( readcount == 1 ) wait(wsem);
signal(x);   signal(rsem);   signal(z);
Reader C.S.
wait(x);
readcount--;
if ( readcount==0 ) signal(wsem);
signal(x);
```

- Writer

```
wait(y);
writecount++;
if ( writecount == 1 ) wait(rsem);
signal(y);
wait(wsem);
Writer C.S.
signal(wsem);
wait(y);
writecount--;
if ( writecount == 0 ) signal(rsem);
signal(y);
```

Λύση με μηνύματα

Reader(i)

```
rmsg = i;
send( readrequest, rmsg );
receive( mbox[i], rmsg );
reader C.S.
rmsg = i;
send( finished, rmsg );
```

Writer(j)

```
rmsg = j;
send( writerequest, rmsg );
receive( mbox[j], rmsg );
writer C.S.
rmsg = j;
send( finished, rmsg );
```

Controller

```
do forever
   if ( count > 0 )
      if ( !empty( finished ) )
         receive( fininshed, msg );
         count++;
      else if ( !empty( writerequest ) )
        receive( writerequest, msg );
        writer_id = msg.id
         count -= 100;
      else if ( !empty( readrequest ) )
         receive( readrequest, msg );
         count--;
         send( msg.id, "OK" );
if ( count == 0 )
   send( writer_id, "OK" );
   receive( finished, msg );
   count = 100;
while ( count < 0 )
   receive( finished, msg );
   count++;
```

Dr. Garmpis Aristogiannis
- EPDO TEI Messolonghi

Εναλλακτική Λύση

- Reader

```
wait(mutex);
if ( writewait > 0 ) { readwait++; signal(mutex);
    wait(readsleep); }
else { readcount++; signal(mutex); }
Reader C.S.
wait(mutex);
readcount--;
if ( readcount==0 && writewait > 0 )
        { writewait--; signal(writesleep); }
else signal(mutex);
```

- Writer

```
wait(mutex);
if ( readcount > 0 )
        { writewait++; signal(mutex); wait(writesleep); }
Writer C.S.
if ( readwait > 0 )
        while ( readwait > 0 )
                { readwait--; readcount++; signal(readsleep); }
        signal(mutex);
else if ( writewait > 0 ) { writewait--; signal(writesleep); }
else signal(mutex);
```