

Operating Systems Exercise 1

2002-05-13

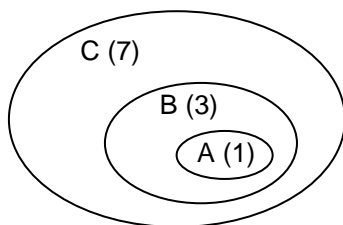
Dominique Emery, s97-602-056

Thomas Bocek, s99-706-319

Philip Iezzi, s99-714-354

Exercise 1.1

	zone A	zone B	zone C
full	0	0	0
empty	1	3	7
mutex	1	1	1



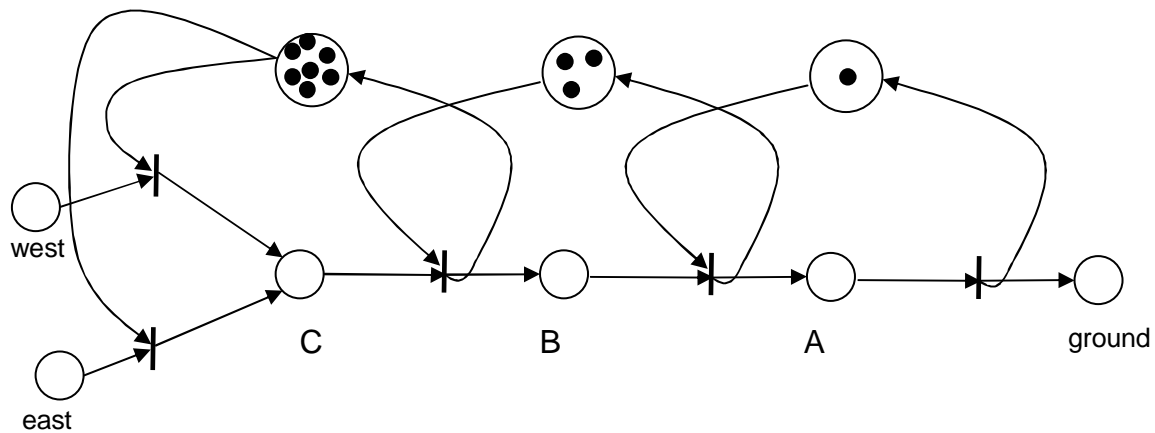
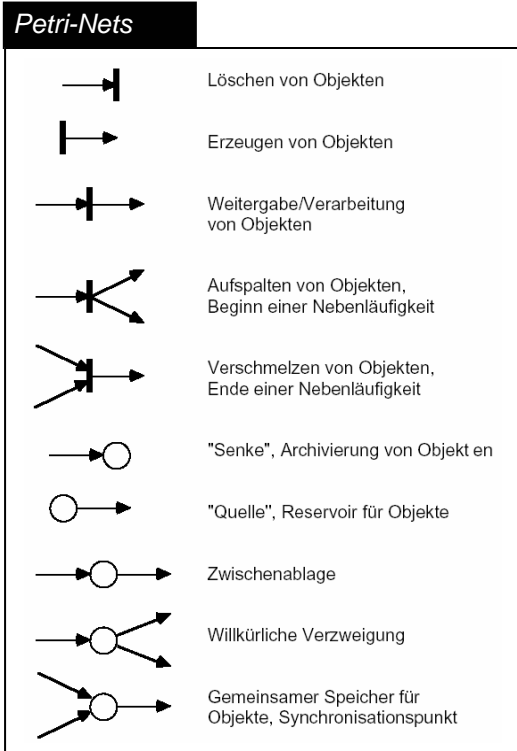
```
// define semaphors
semaphor A_full, A_empty, A_mutex
semaphor B_full, B_empty, B_mutex
semaphor C_full, C_empty, C_mutex
```

```
// initialization
```

```
A_full := 0
A_empty := 1
A_mutex := 1
B_full := 0
B_empty := 3
B_mutex := 1
C_full := 0
C_empty := 7
C_mutex := 1
```

```
begin
  // enter area C
  P(C_empty)
  P(C_mutex)
  move_into_C
  V(C_mutex)
  V(C_full)
  // move from C to B
  P(C_full)
  P(C_mutex)
  move_out_of_C
  // enter area B
  P(B_empty)
  P(B_mutex)
  move_into_B
  V(B_mutex)
  V(B_full)
  V(C_mutex)
  V(C_empty)
  // move from B to A
  P(B_full)
  P(B_mutex)
  move_out_of_B
  // enter area A
  P(A_empty)
  P(A_mutex)
  move_into_A
  V(A_mutex)
  V(A_full)
  V(B_mutex)
  V(B_empty)
end
```

Exercise 1.2



Exercise 1.3

sem $S_x = 0, S_y = 0$

A	B	C
Action	P(S_x)	P(S_y)
V(S_x)	Action	Action
...	V(S_y)	...
	...	

-> A -> B -> C

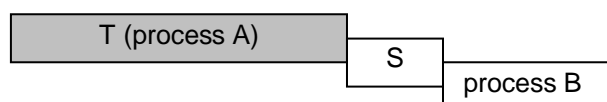
If all processes are started at once, process a comes first, then b and then c.

Exercise 1.4

The WAITUNTIL scheme is not used out of efficiency reasons. In the implementation of the WAITUNTIL scheme we would be forced to recheck the boolean expression after every assignment statement that could affect its value. There's no need to do this in the Hoare and Brinch Hanson scheme.

Exercise 1.5

e.g.:



efficiency

Here, efficiency is understood as the ratio of "useful time" to the total time.

"useful time" \approx service time
total time \approx overhead time

$$CPU - efficiency = \frac{T}{T + n \cdot S}$$

where n is the amount of context switches

a) $Q = \infty$

The overhead time is exactly one context switch ($n=1$).

$$\frac{T}{T + n \cdot S} = \frac{T}{T + S}$$

b) $Q > T$

same as a), overhead time is exactly 1 context switch

$$\frac{T}{T + n \cdot S} = \frac{T}{T + S}$$

c) $S < Q < T$

The process will require T/Q number of context switches

$$\frac{T}{T + n \cdot S} = \frac{T}{T + \frac{T}{Q} \cdot S} = \frac{1}{1 + \frac{S}{Q}} = \frac{Q}{Q + S}$$

d) $Q = S$

$$\frac{T}{T + n \cdot S} = \frac{T}{T + \frac{T}{Q} \cdot S} = \frac{T}{T + \frac{T}{S} \cdot S} = \frac{1}{2}$$

e) Q close to 0

$$\lim_{Q \rightarrow 0} \frac{T}{T + \frac{T}{Q} \cdot S} = 0$$

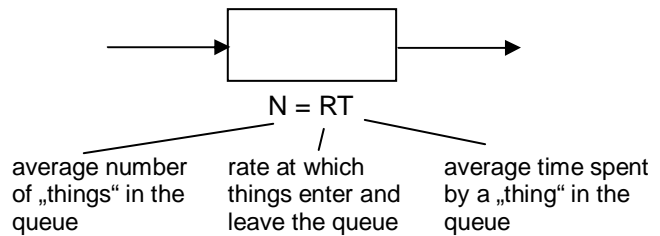
This means that the overhead (context switching) dominates the use of the CPU, so no "useful work" is being performed.

Exercise 1.6

Little's Law

$$N = RT$$

$$\text{avg_number_tasks} = \text{avg_resp_time} * \text{avg_throughput}$$



Proof:

- W accumulated time within system
- T time interval
- C completed tasks

$$N = R \cdot T$$

$$\frac{W}{T} = \frac{C}{T} \cdot \frac{W}{C}$$

This formula is always true in an arbitrary, but fixed observation period as we can reduce this to $W/T = W/T, 1 = 1$