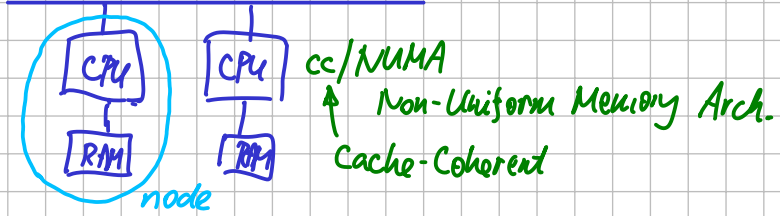
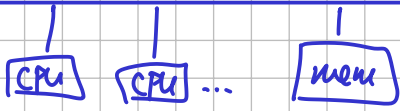
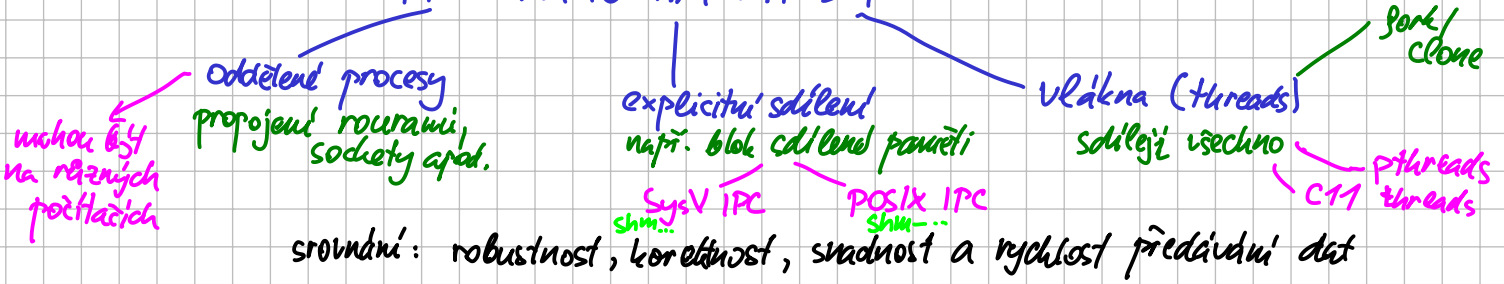


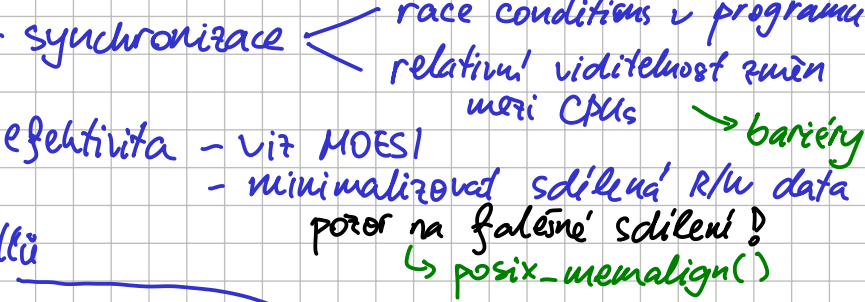
# SMP



## PROGRAMOVÁNÍ NA SMP



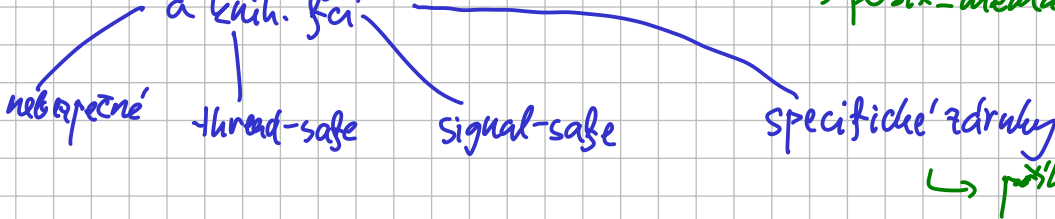
problémy se sdílením



```

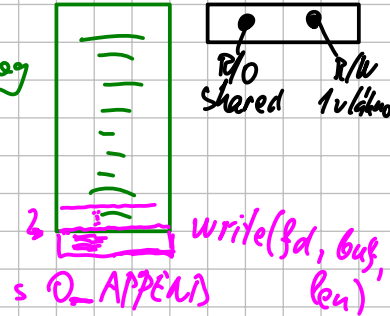
c++
↓
r[cpu]
r++
[cpu] ← r
struct state;
3 ... current;
    
```

bezpečnost syscallů a knih. fci

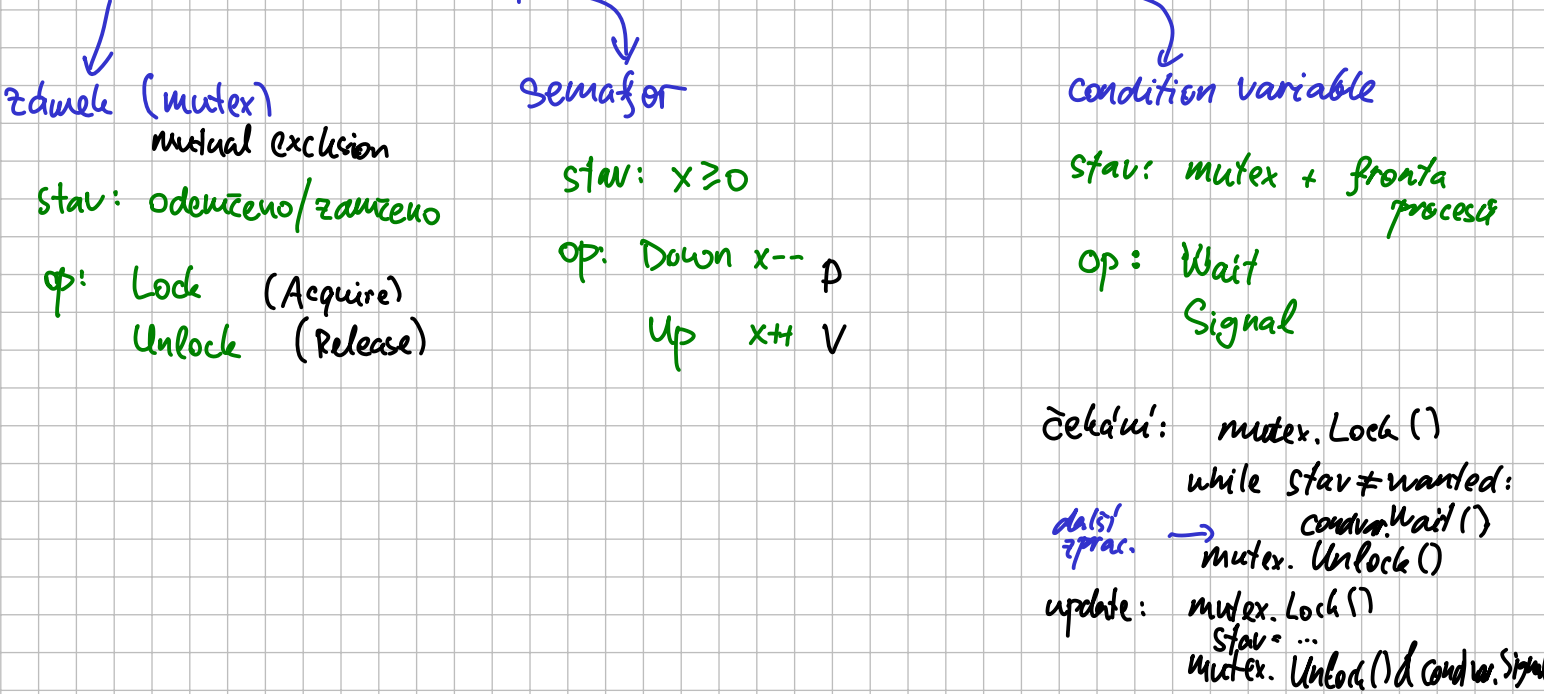


### Thread-Local Variables

- pthread\_getspecific - otravné & pomalé
  - GCC: --thread
  - C11: -Thread-local  
↳ thread-local
- storage class (+static/extern)  
implementace pomocí GS



## SYNCHRONIZAČNÍ PRIMITIVA



Příklad: fronta → seznam... → mutex <sup>M</sup>: chrdlu užití stav  
 → semafor <sup>S</sup>: počítá prvky

Enqueue(x):

M. Lock  
 přidáme x do seznamu  
 M. Unlock  
 S. Up

Dequeue:

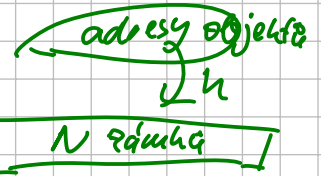
S. Down  
 M. Lock  
 odpojíme prvek ze seznamu  
 M. Unlock

Invariant: délka seznamu = semafor + # rezervací  
 ↑  
 když není zavazeny M

Dále:

rychlost sync. primitiv — na Linuxu futex — ale: zámky jsou sdílené r/w data !  
 (typ. případ user-space, jinak kernel)

granularita: prvky ↔ globální  
 nešerdné zámky



AB-BA deadlock

