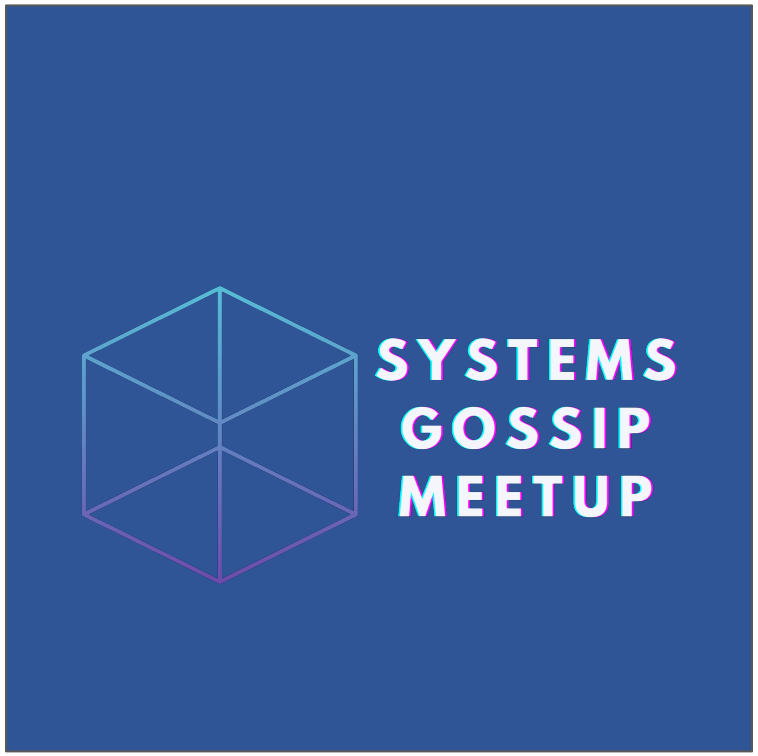


Learning large systems using peer-to-peer gossip

Policy Against Harassment at ACM Activities

OS Meetup wants to encourage and preserve this open exchange of ideas, which requires an environment that enables all to participate without fear of personal harassment. We define harassment to include specific unacceptable factors and behaviors listed in the ACM's policy against harassment. Unacceptable behavior will not be tolerated.
https://www.acm.org/about-acm/policy-against-harassment

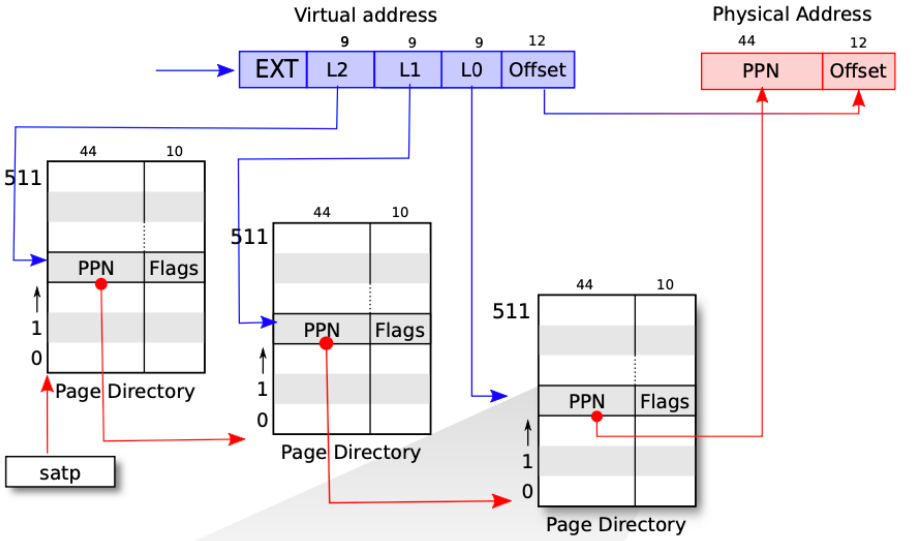
Page Fault

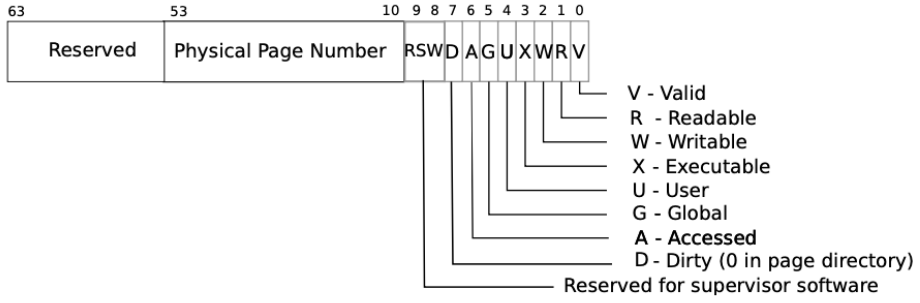
- Recall:
  - Page table
- Page allocation
  - fork
- Share memory between parent and child processes?
  - Introduction
  - Problems?
- Fork and Copy-on-Write
  - Benefit
- Page Fault
  - Lazy allocation
  - ~~Demand paging~~

Recall: Trap

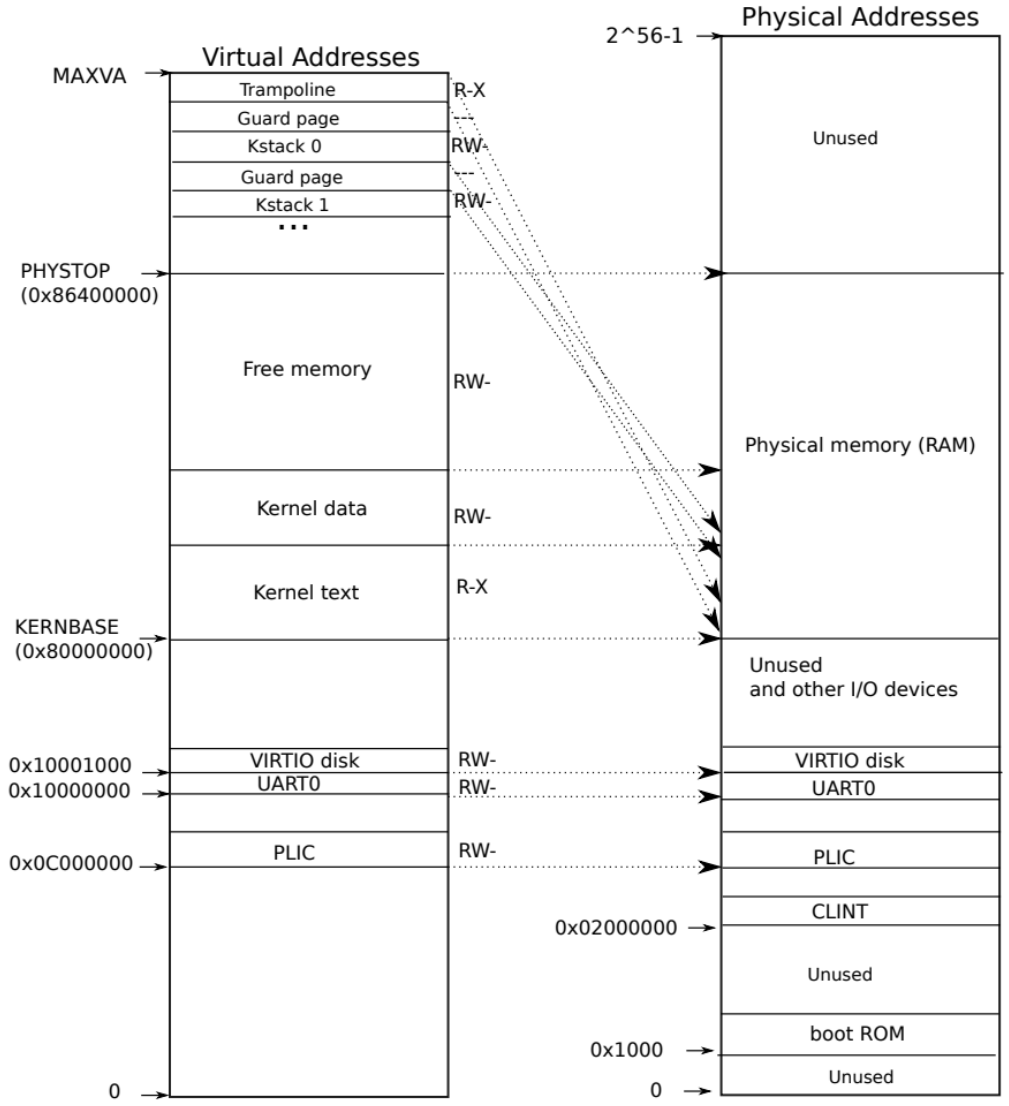
```
132  void
133  syscall(void)
134  {
135    int num;
136    struct proc *p = myproc();
137
138    num = p->trapframe->a7;
139    if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
140      p->trapframe->a0 = syscalls[num]();
141    } else {
142      printf("%d %s: unknown sys call %d\n",
143              p->pid, p->name, num);
144      p->trapframe->a0 = -1;
145    }
146  }
147
```

Recall: Page Table





Recall: Page Table



Recall: Page Table

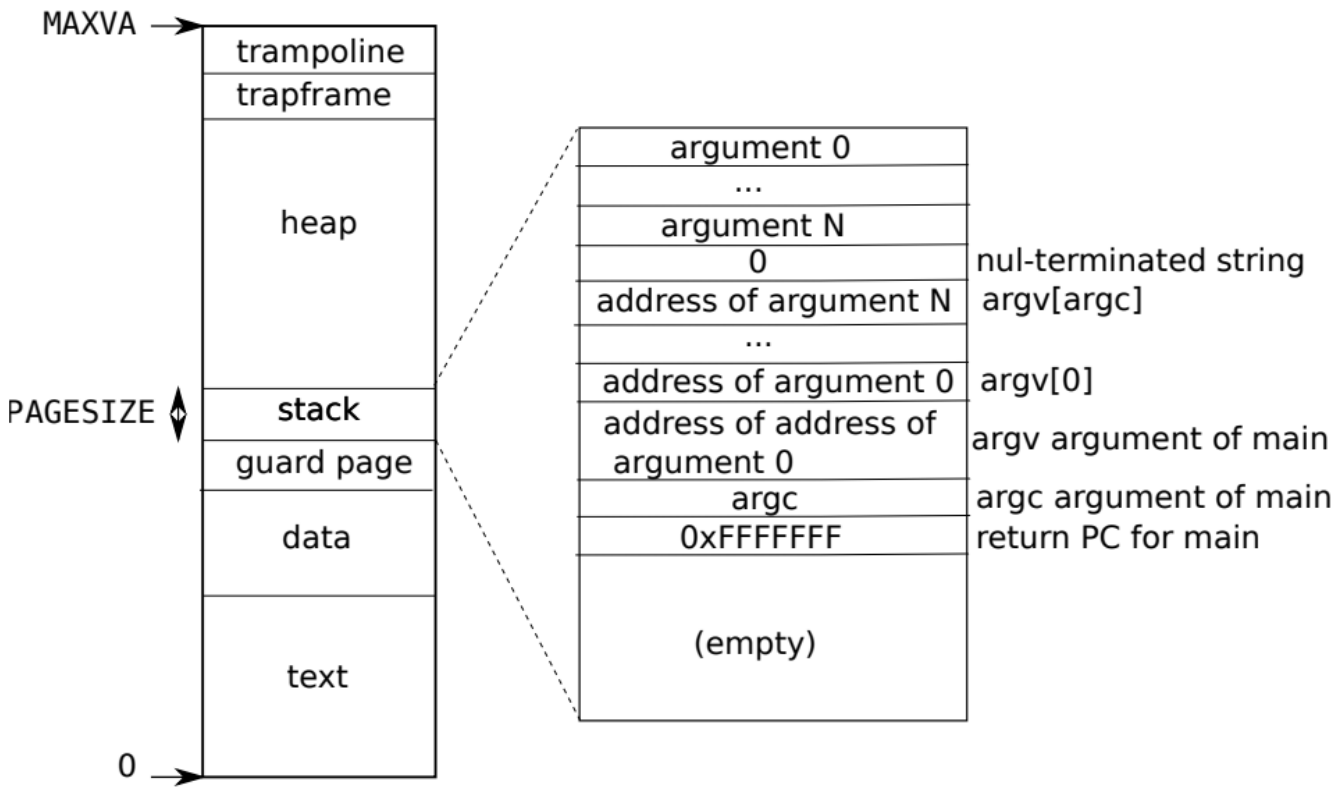
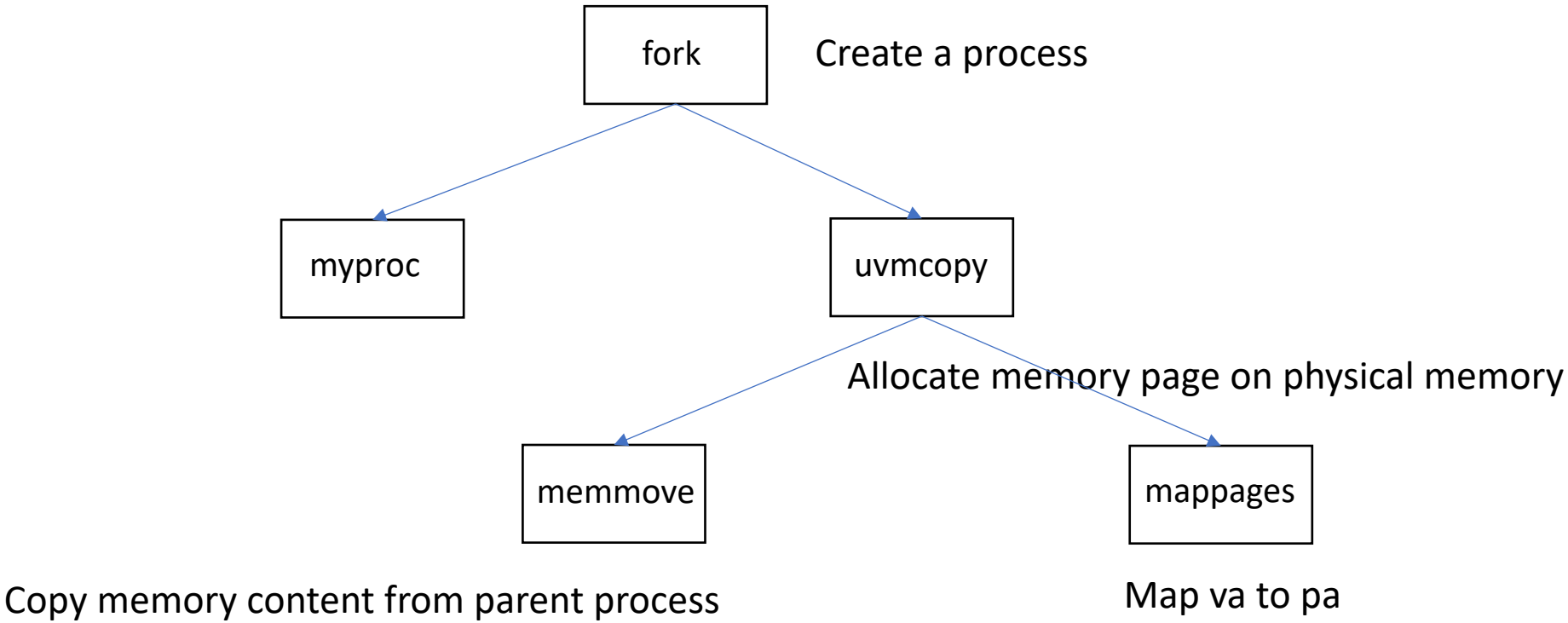
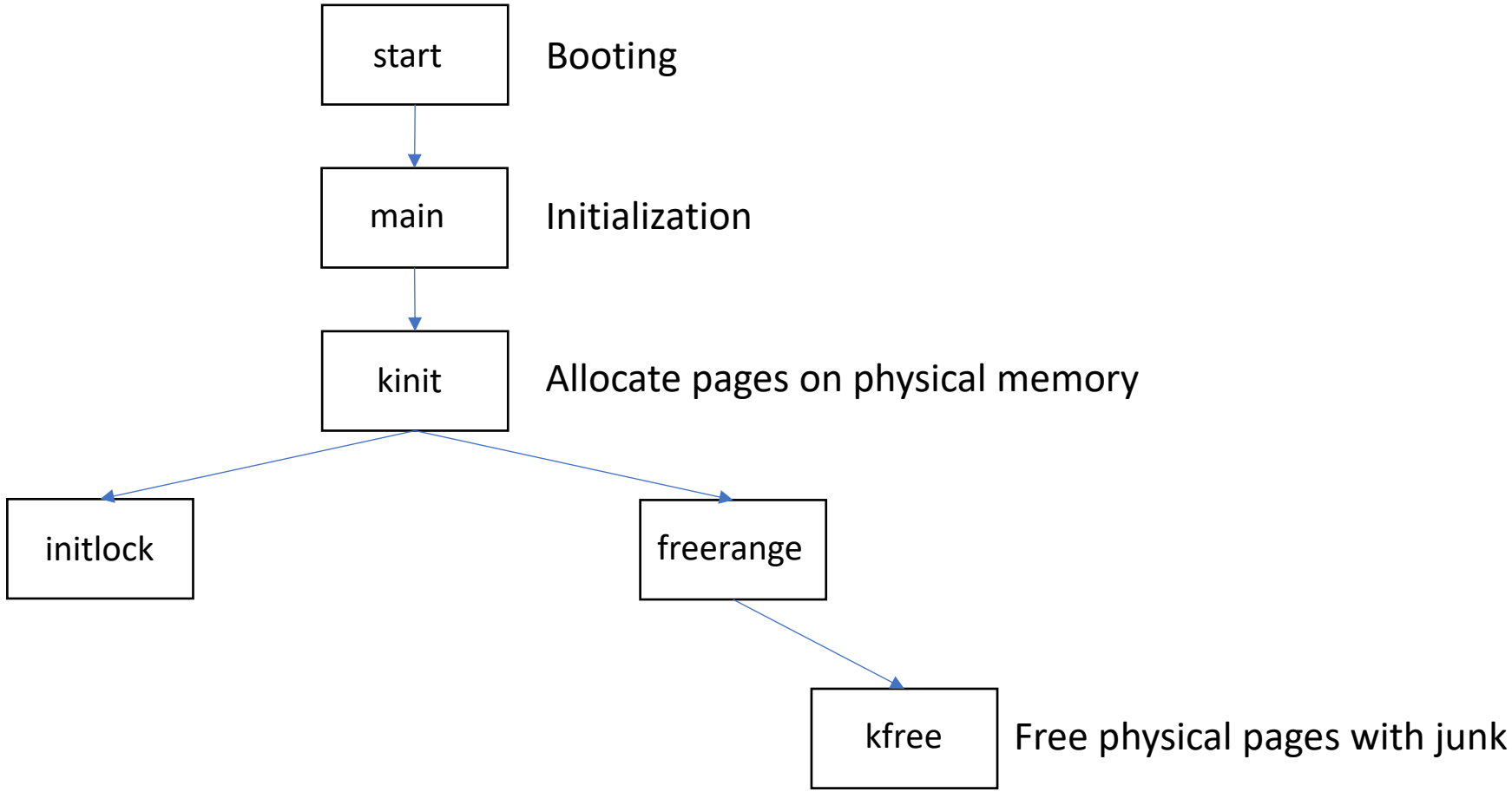


Figure 3.4: A process's user address space, with its initial stack.

Page Allocation: fork

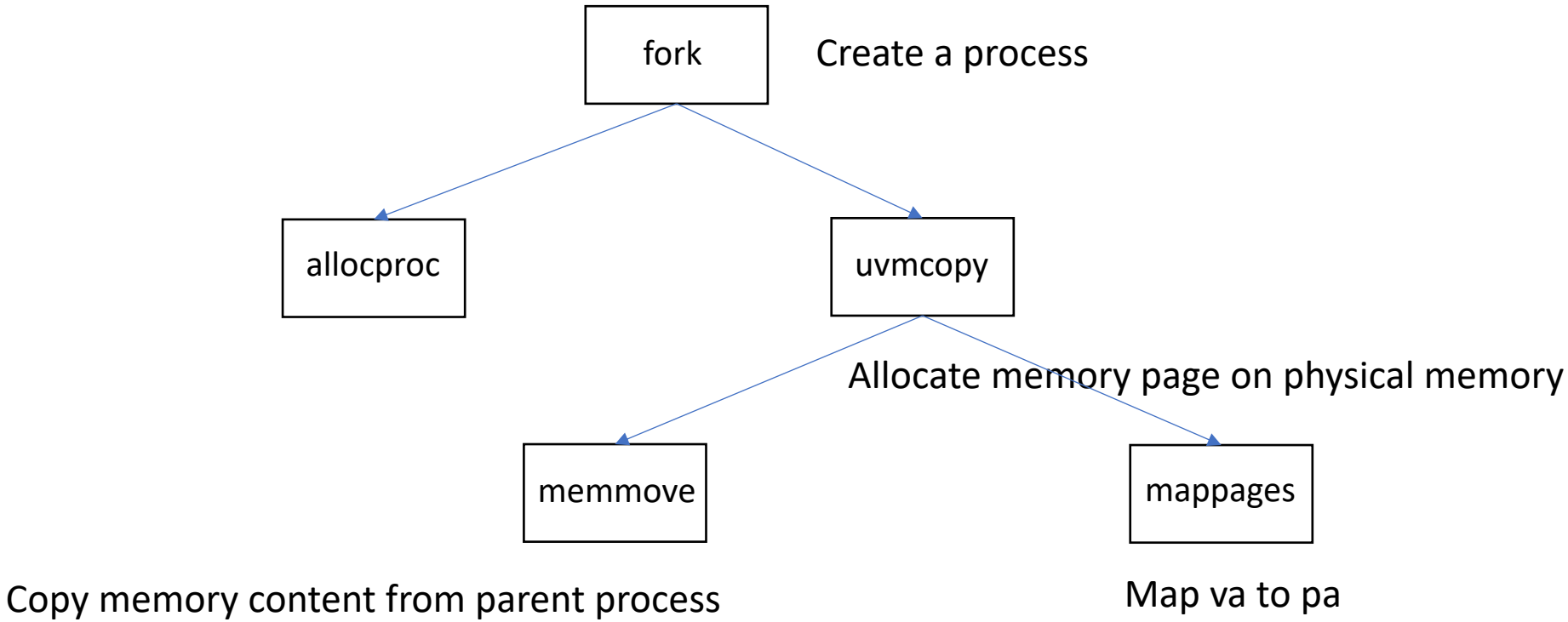


Page Allocation: free pages



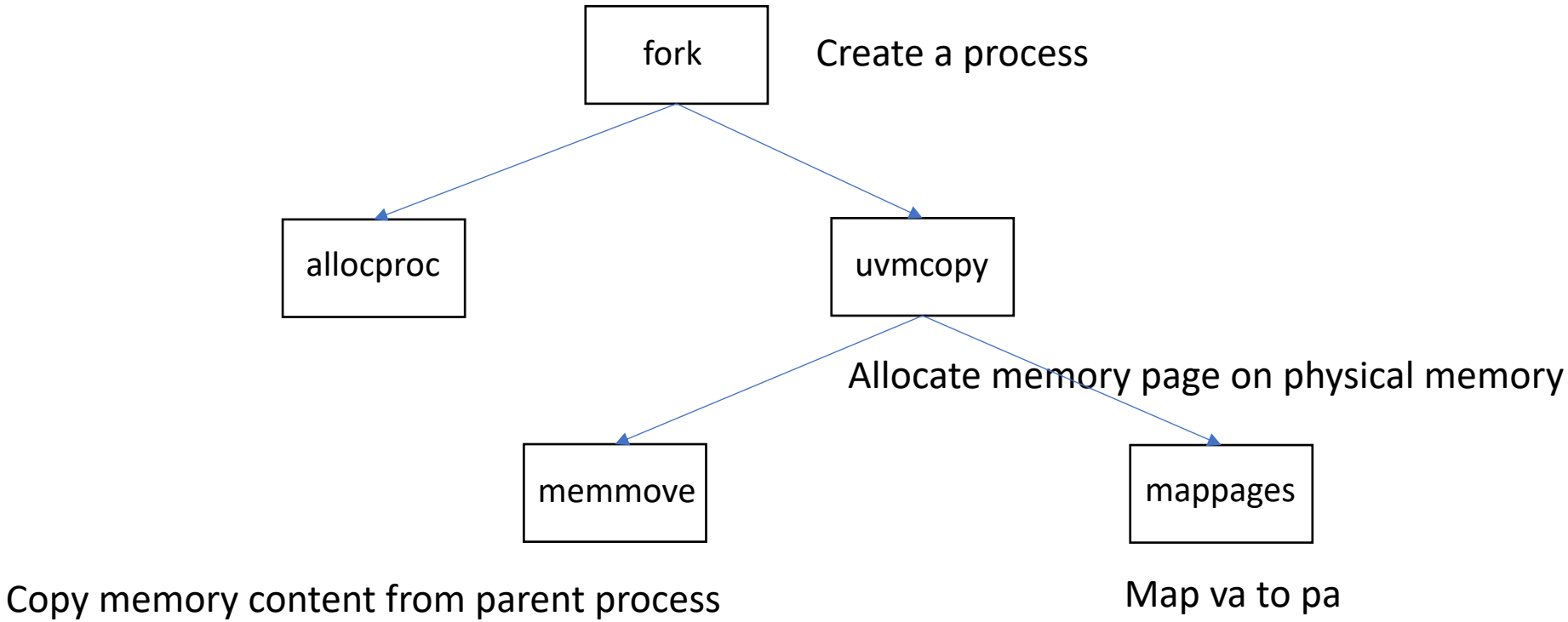
Open Question: Process

Do we have a root process in OS?

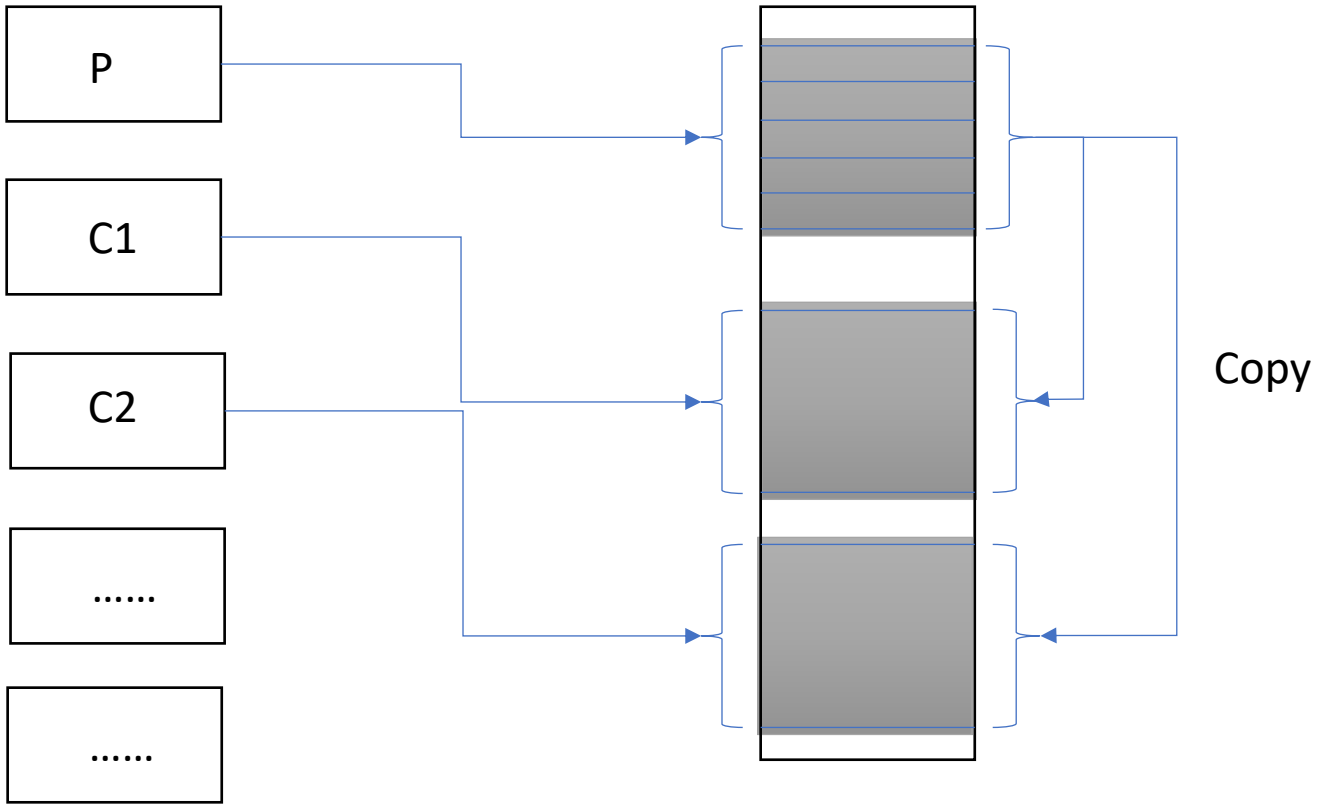


Open Question: Process

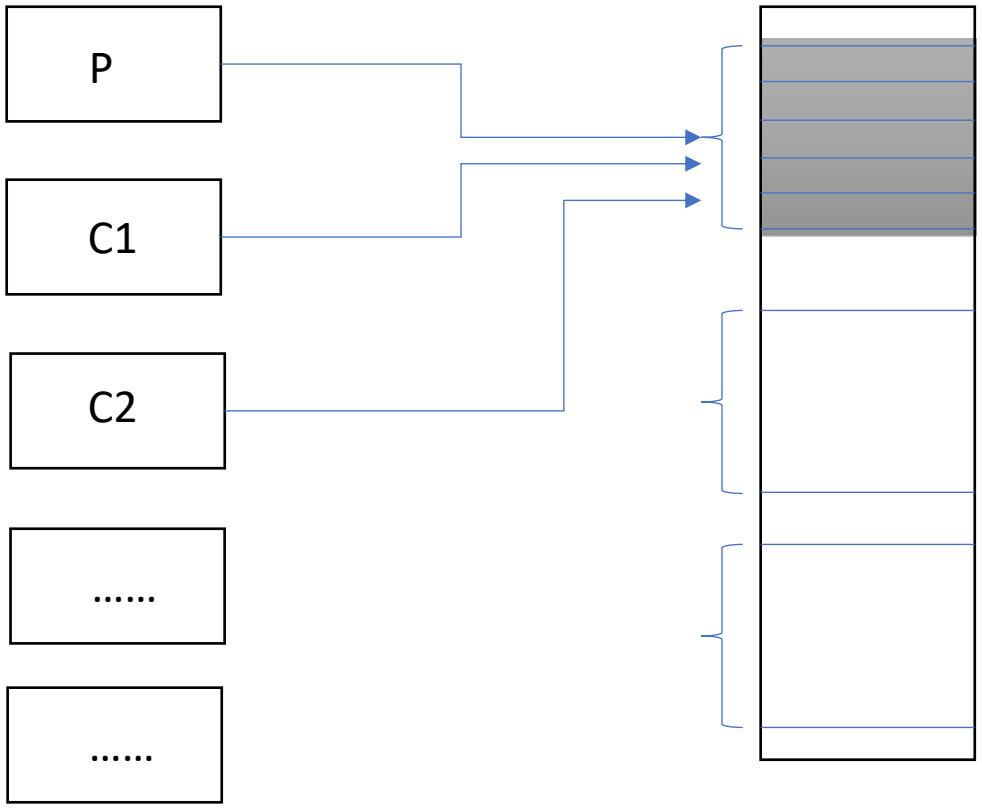
Where does OS store the state data of each process?



fork: Not Memory Efficient

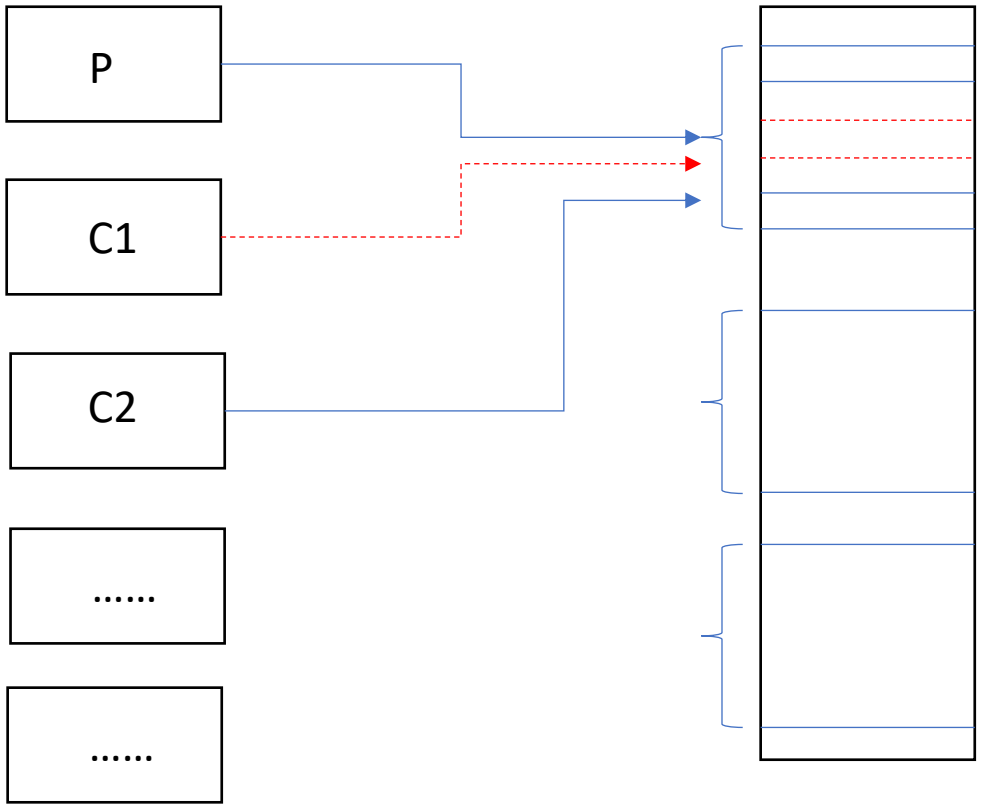


fork: Share Memory

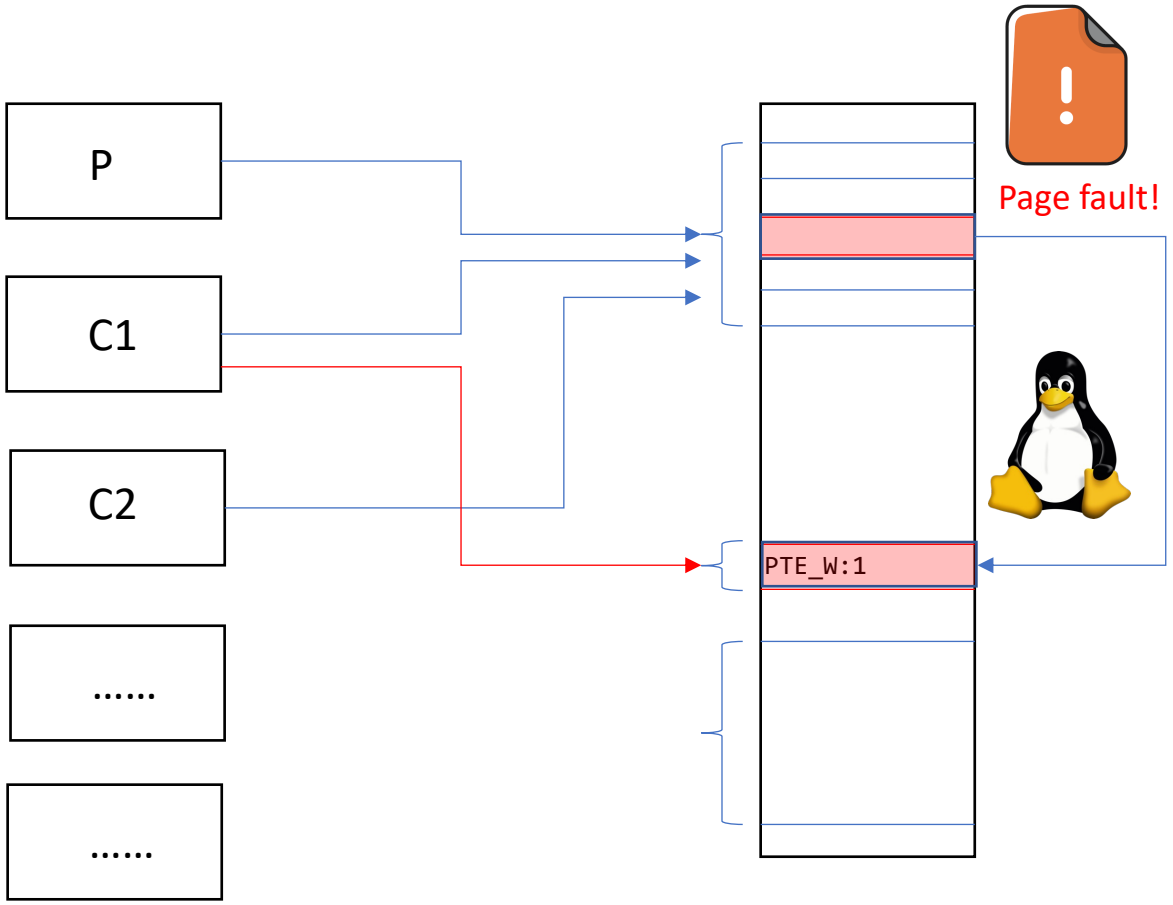


Share Memory: Problem

Operation system isolation broken?

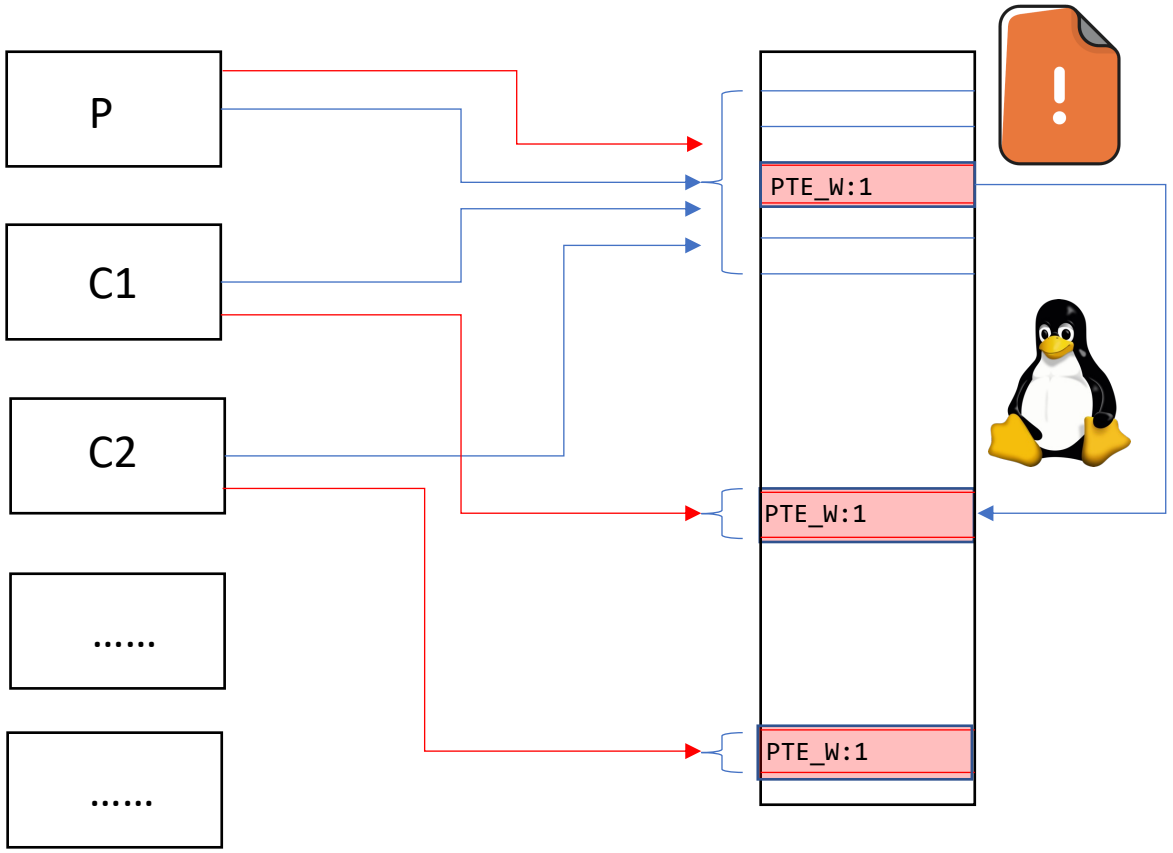


Share Memory: Copy-on-Write



Open Discussion: Copy-on-Write

Do we need to copy for other child processes?



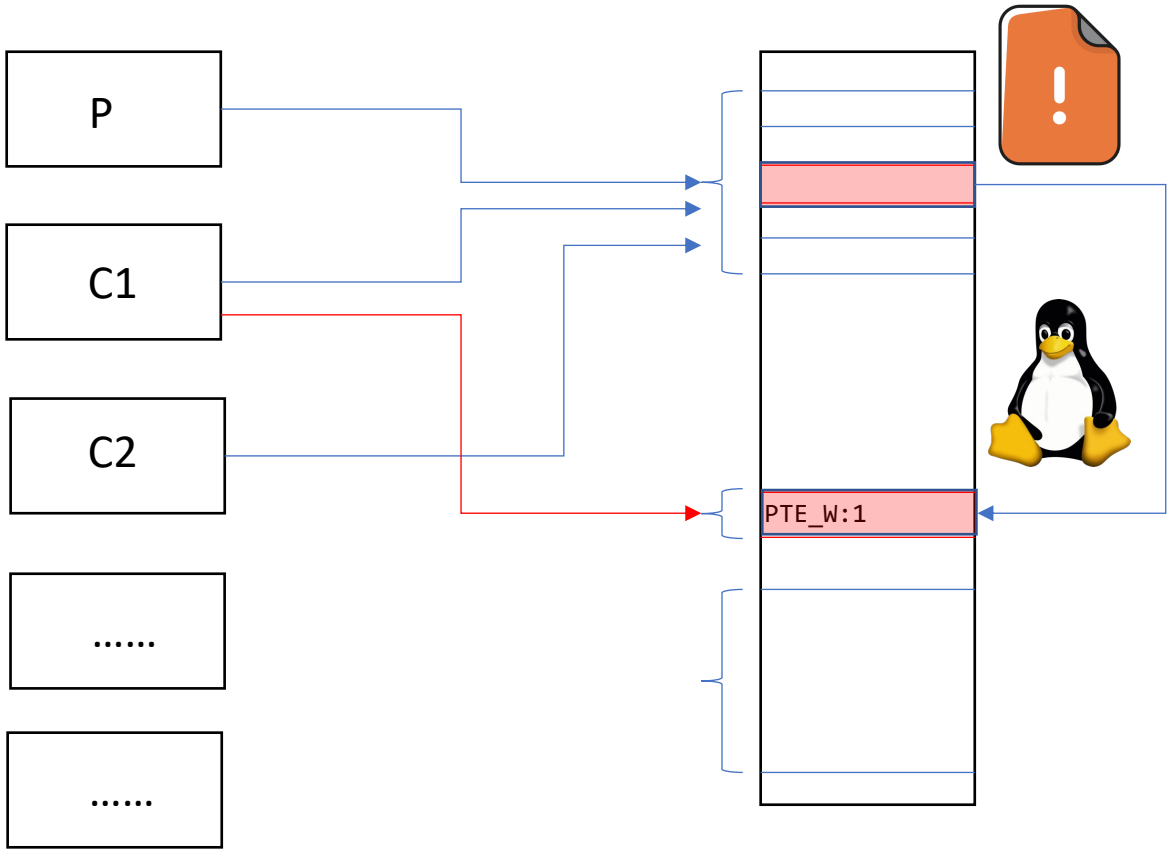
Copy-on-Write: Benefit & Drawback

- Benefit:
  - Memory saving
  - Quick start on process creating

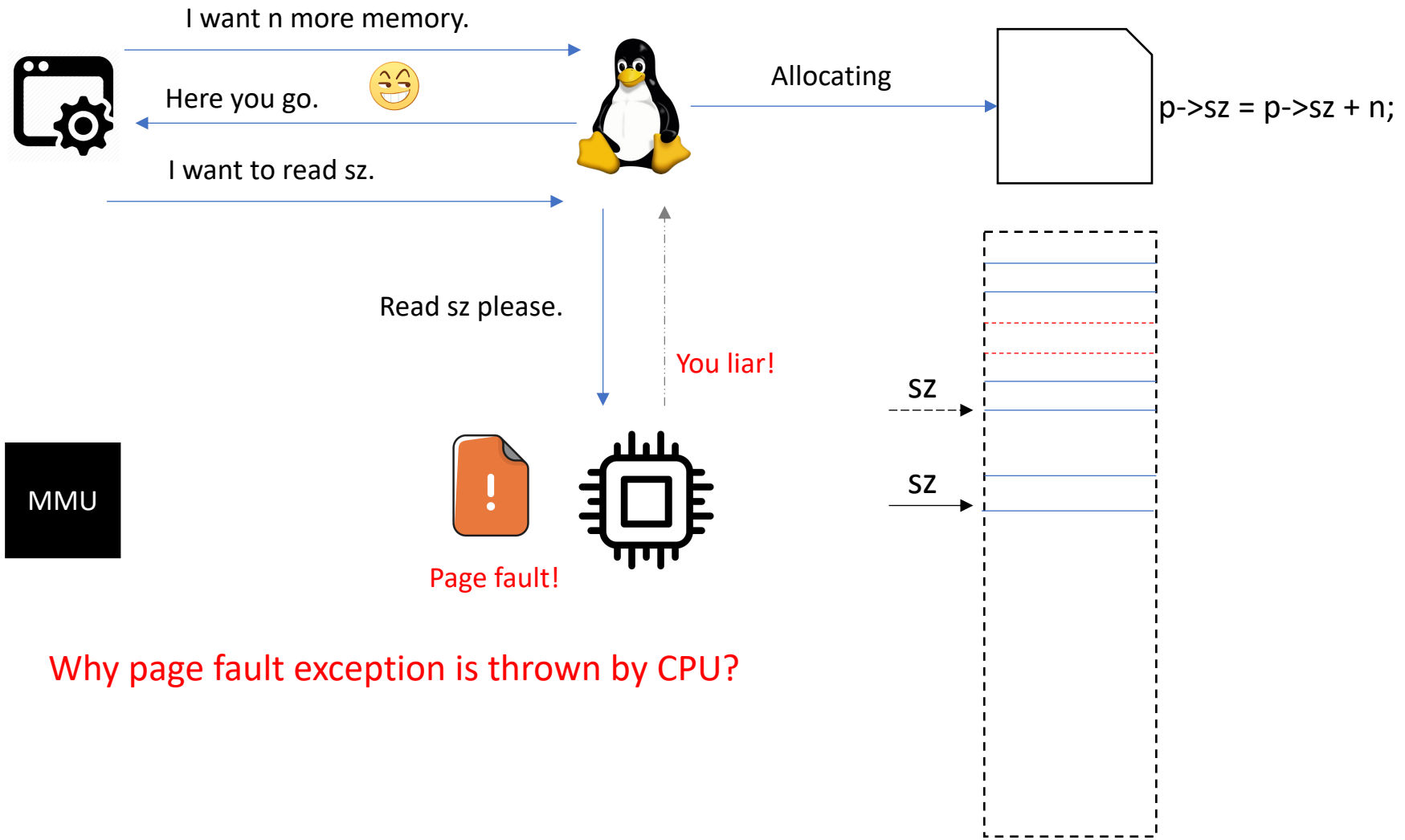
- Drawback:
  - Book keeping for page free

Discussion: fork in xv6

fork in xv6 is not implemented with Copy-on-Write.

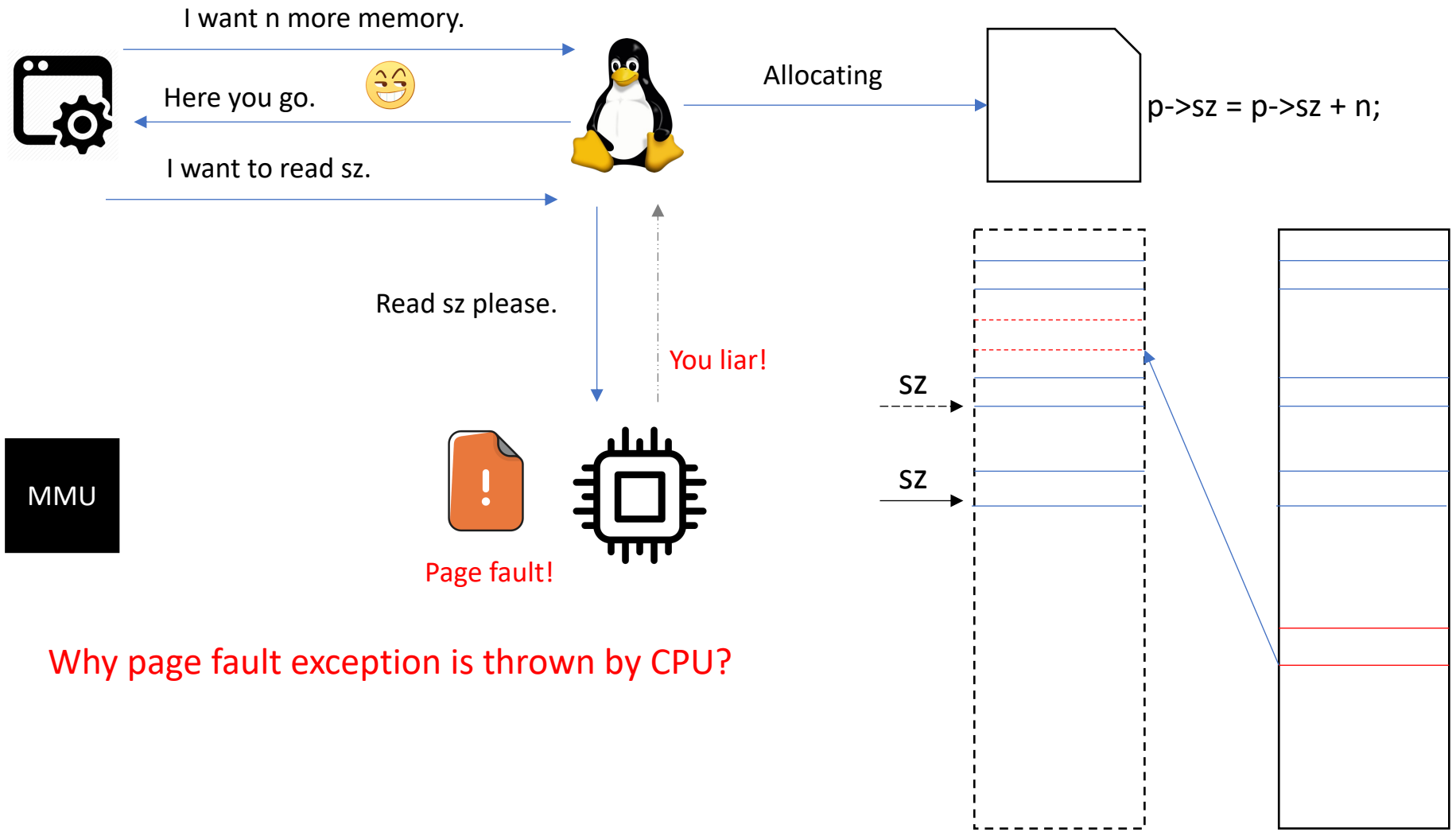


Page Fault: Lazy Allocation



Why page fault exception is thrown by CPU?

Page Fault: Lazy Allocation



Why page fault exception is thrown by CPU?

Summary

- Page table
- Page allocation: fork
- Memory sharing between parent and child process
- Copy on write: fork
- Page fault: lazy allocation
- Next
  - Hongwang Li on Intrrupts