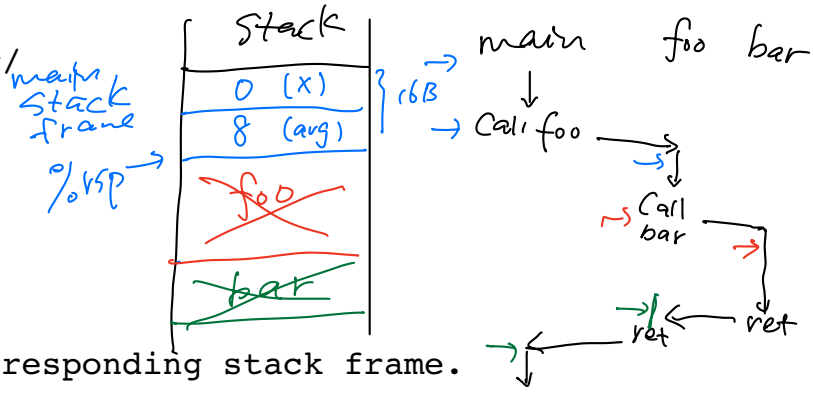
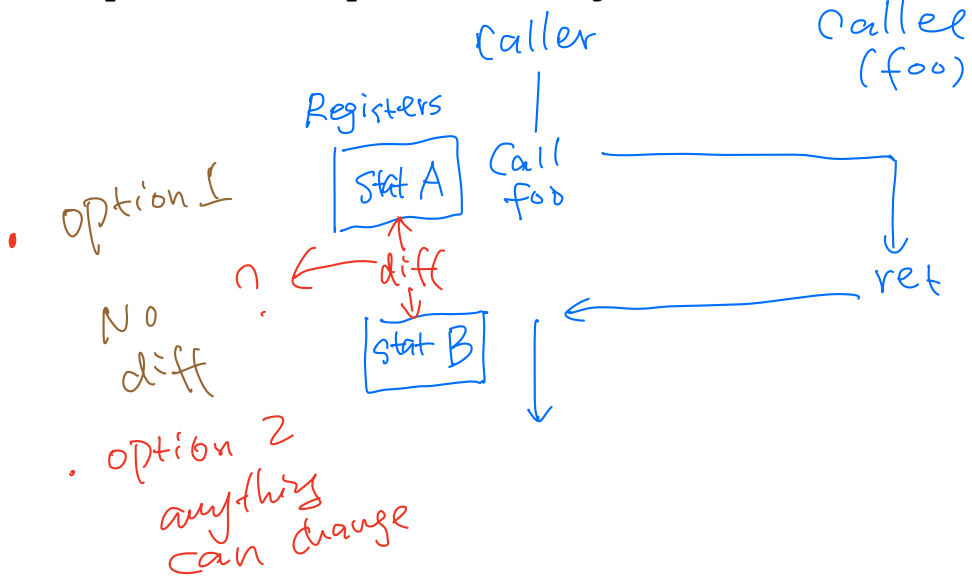
Week 5.b
CS3650
02/07 2024
https://naizhengtan.github.io/24spring/

1. OS scheduling
2. Threads
3. Intro to concurrency

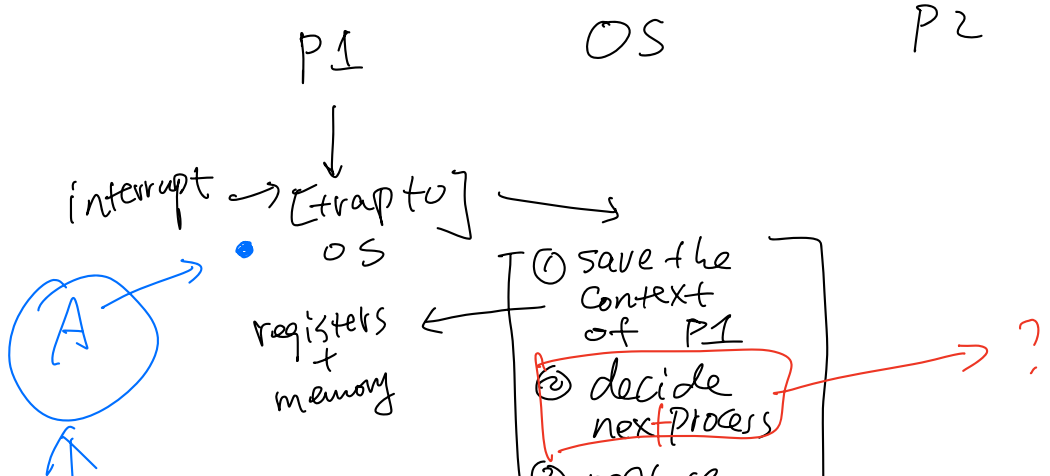


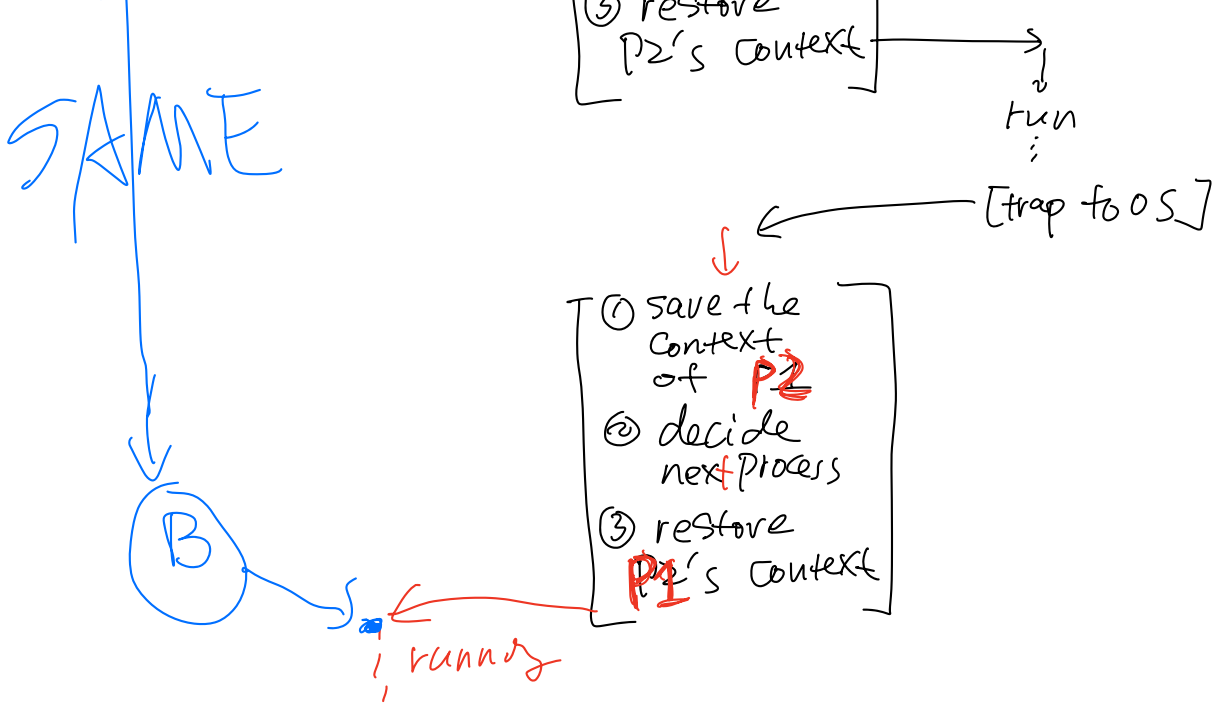
Recap, True or False?

- T each function invocation has a corresponding stack frame.
- T all local variables are located on the stack frames.
- T the first four function arguments are stored on registers.
- T the return value is stored on registers. %rax
- T call-preserved registers are registers that need to be saved by callee
- T caller does not have to save call-clobbered registers, if their values are not useful
- T %rax is a call-clobbered register
- T %rsp is a call-preserved register

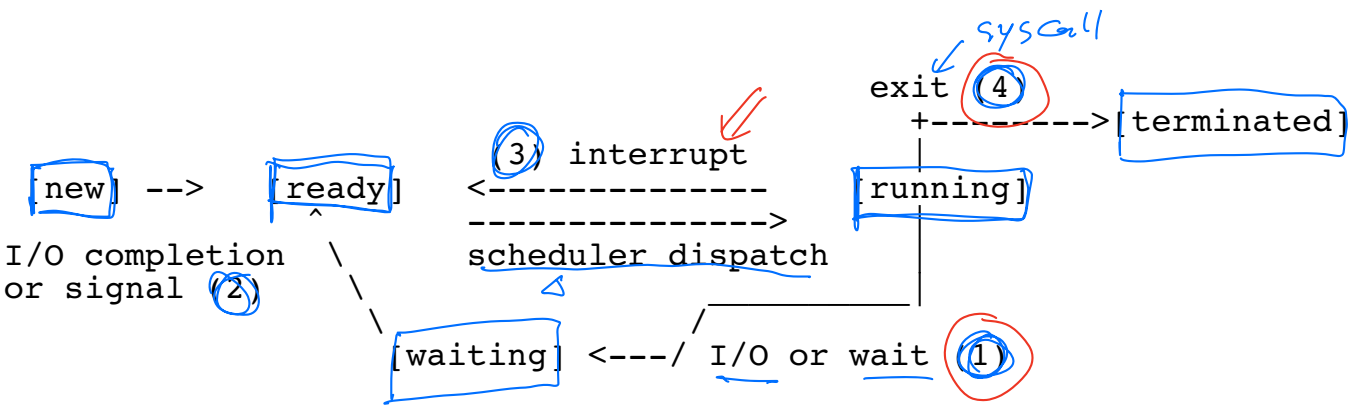


- context switch





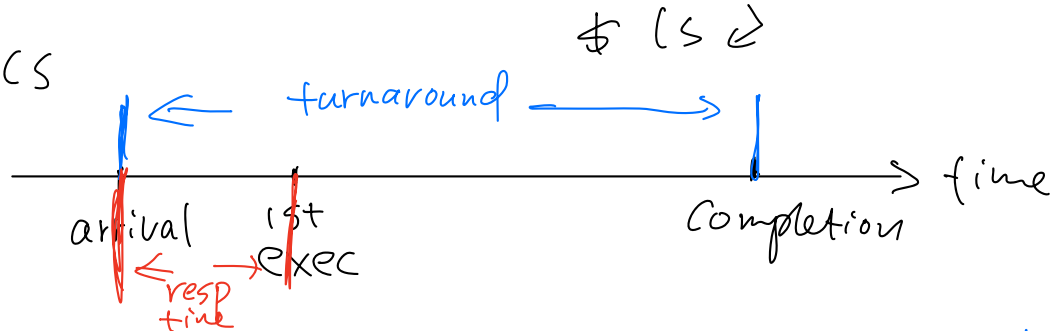
- OS scheduling



preemptive : (1) - (4)

non preemptive : (1), (4)

- metrics



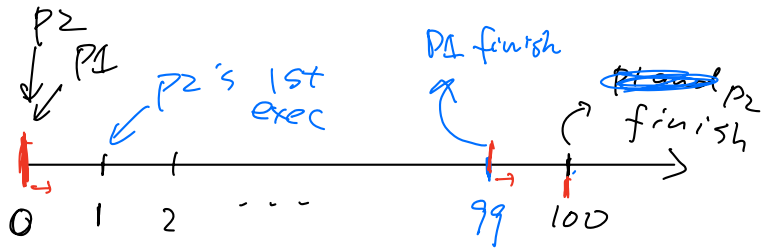
turnaround time = |completion − arrival|

response time = |1st exec − arrival|

fairness = no starvation

A scheduling problem:
(with slice of 1 unit of time)

process	arrival	running
P1	0	50
P2	0	50



Schedule:

P1, P1 ··· P1, P2 ··· P2

P1, P2, P1, P2 ··· P1, P2 ⇒ RR : round-robin

Question: average turnaround time:

$$\frac{(99-0)+(100-0)}{2} = 99.5$$

Question: average response time:

$$\frac{0+1}{2} = 0.5$$

- Threads

interfaces: tid thread_create(func_ptr, void *arg); (function)

void thread_exit();

void thread_wait(tid);

```
int a
Void foo() {... a=1 }
Void bar() {... a->1 }

int main() {
    thread_create(foo, NULL);
    thread_create(bar, NULL);
```

3 threads
- main
- foo

}

bar

Q: processes vs. threads