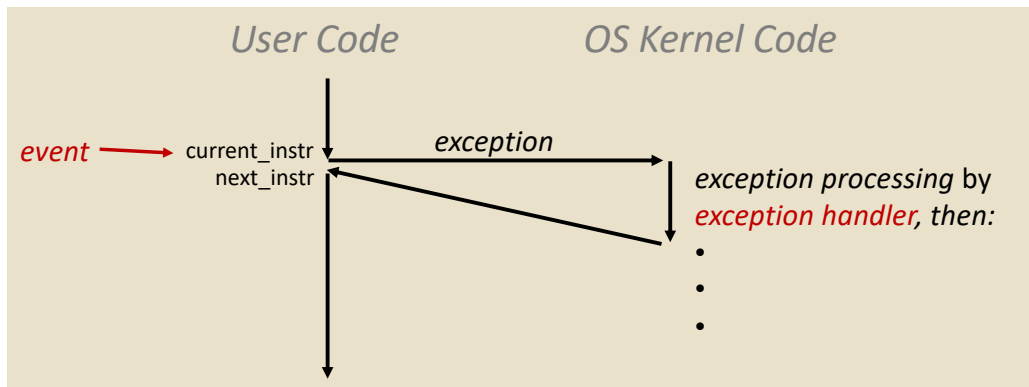


# Exceptions: Handout

- ❖ An *exception* is transfer of control to the operating system (OS) kernel in response to some *event* (i.e. change in processor state)
  - Kernel is the memory-resident part of the OS
  - Examples: division by 0, page fault, I/O request completes, Ctrl-C

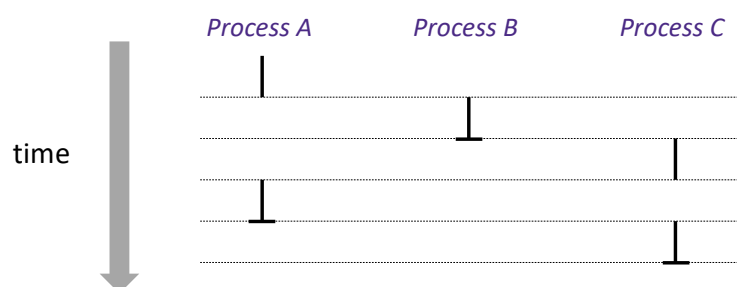


- ❖ *How does the system know where to jump to in the OS?*

## Concurrent Processes: Handout

Assume only one CPU

- ❖ Each process is a logical control flow
- ❖ Two processes *run concurrently* (are concurrent) if their instruction executions (flows) overlap in time
  - Otherwise, they are *sequential*
- ❖ Example: (running on single core)



# Fork Example: Handout

```
void fork1() {
    int x = 1;
    pid_t fork_ret = fork();
    if (fork_ret == 0)
        printf("Child has x = %d\n", ++x);
    else
        printf("Parent has x = %d\n", --x);
    printf("Bye from process %d with x = %d\n", getpid(), x);
}
```

63

## Polling Question

❖ Are the following sequences of outputs possible?

■ Vote at <https://PollEv.com/wolfson>

```
void nestedfork() {
    printf("L0\n");
    if (fork() == 0) {
        printf("L1\n");
        if (fork() == 0) {
            printf("L2\n");
        }
    }
    printf("Bye\n");
}
```

Seq 1:      Seq 2:

L0	L0
L1	Bye
Bye	L1
Bye	L2
Bye	Bye
L2	Bye

- |    |               |     |
|----|---------------|-----|
| A. | No            | No  |
| B. | No            | Yes |
| C. | Yes           | No  |
| D. | Yes           | Yes |
| E. | We're lost... |     |

64