CS 61 Lecture 23 Nov 19

Timed Wait

Parent forks child

Parent waits until { child exits / 3/4 second } whichever happens first

Polling: Process waits for event while runnable

+ easy to wait for complex conditions

− poor utilization

Blocking: Process waits for event while not runnable; becomes runnable when event occurs

+ good utilization

− difficult to express complex conditions

Signal: Process-level abstraction of HW interrupt

① Blocking system calls returns early when signal delivered. Syscall returns -1 and errno = EINTR

② Process can install their own signal handlers

SIGCHLD: delivered to parent when child exits.

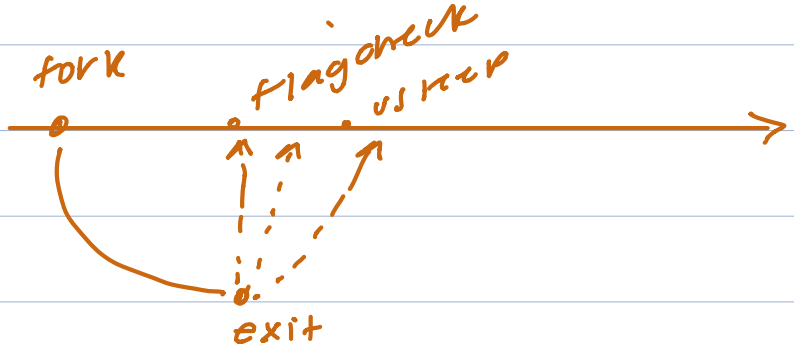
parent blocks waiting for timeout, SIGCHLD interrupts

SIGALRM: delivered after seconds.

sigalarm interrupts waitpid

code: progress4

FORK vfork



void signal

:

```
if (!flag)
    r = vfork( ... );
    void(r)
```

code: timedwait05-selfpipe

Problem Set

control-C

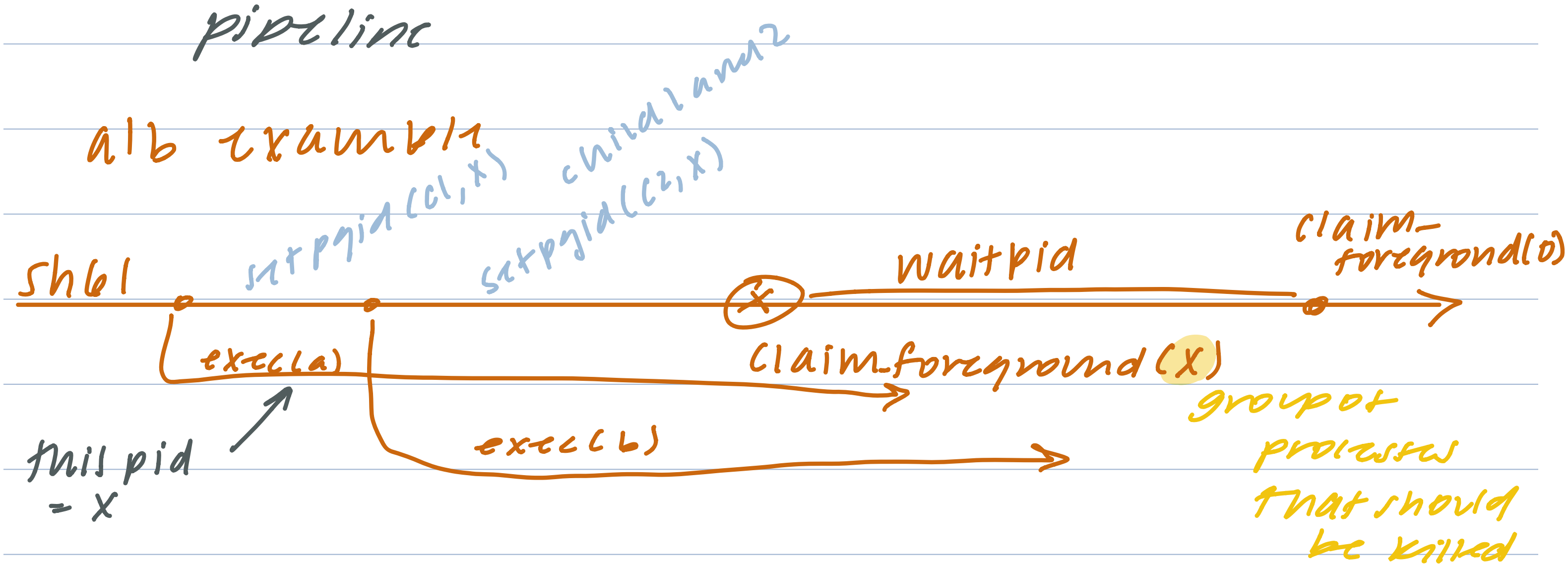
* pstree *

2nd to last part of pset:

interruption in shell

- kills all processes in currently foregrounded pipeline

a|b example



X is a process group id

only for commands in foreground

- exec(a) and exec(b) need to change their process group id to x before running

setpgid(0, X)