Job control

- Send SIGSTOP signal to freeze a process
- Send SIGCONT signal to resume a process

On command line:

```
kill -SIGSTOP pid
```

Programmatically:

```
kill (pid, SIGSTOP/CONT);
```
waitpid (pid_t pid, int *status, int flags);

-1 if you don't care to be specific

child you want to get info about, or
pointer to a collection of bits that tells you into abut the child

D: default: wait for child to terminate
WUNTRACED:
wait for child to stop
WCONTINUED:
wait for child to continue

Execvp:
runs an executable on the machine

int execvp(const char *executable,
            char ** const argv[]);

what we might want: "make", "clean", NULL

pid 10

"variables"

pid 11

"variables"
what actually happens

DS throws out virtual mem, installs new heap stack / text segment
Pipes

Special file in memory with a write end that you can write to, and a read end that echoes those bytes.

```c
write(f, "hi!");
```

FD: 0 1 2 3 4

OF: mode: R mode: W

Variable: pipe

write is right

Learn to read before you learn to write
Read:
  - If nothing is in the pipe, waits for at least 1 byte to be available.

  - If all write file descriptors are closed, no more data can ever go into the pipe, returns 0.