

CS 208

Mon, 15 May 2023

fd table

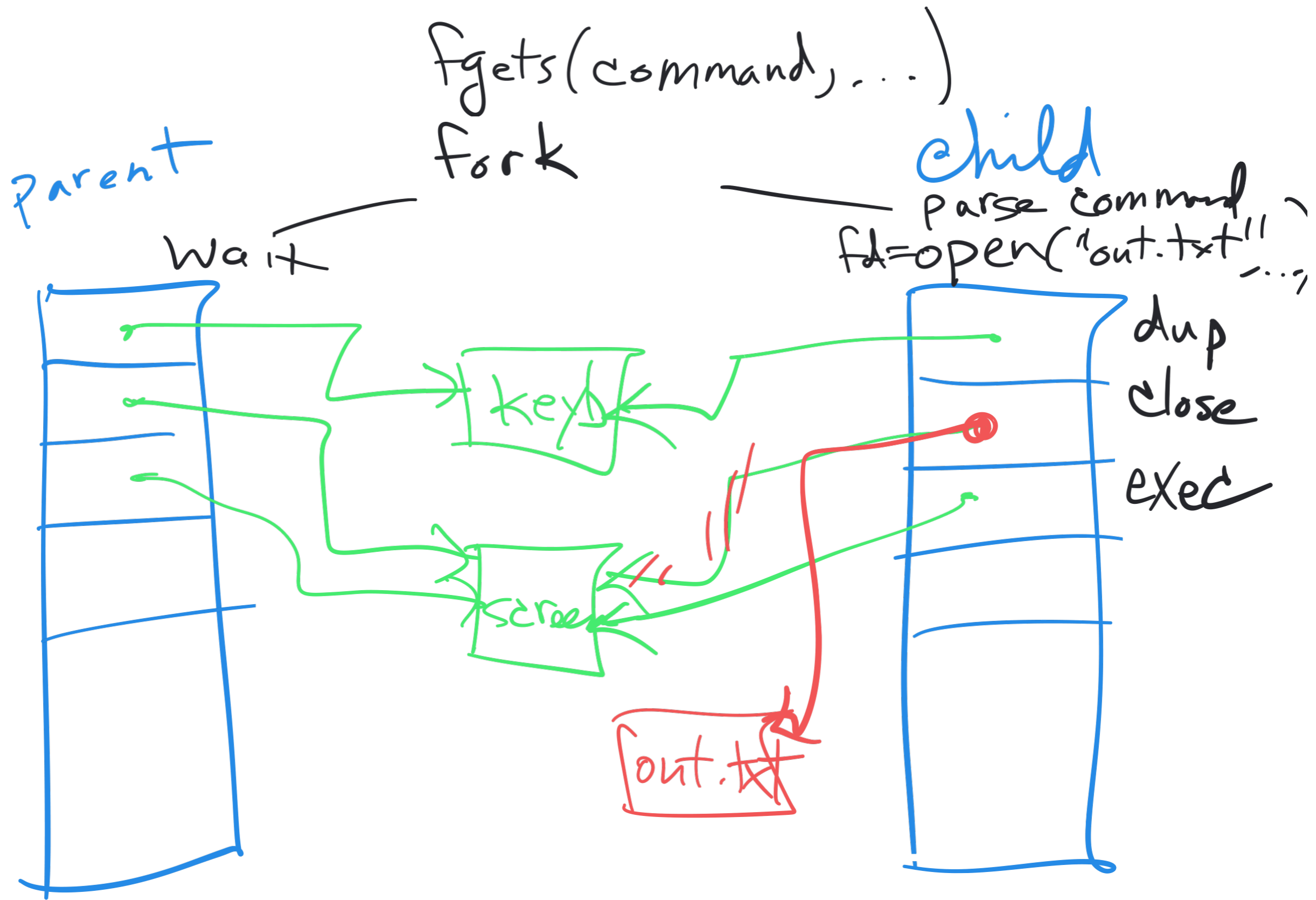


```
fd = open("elk.txt",  
          O_RDONLY);
```

```
dup2(fd, 0);
```

```
close(fd);
```

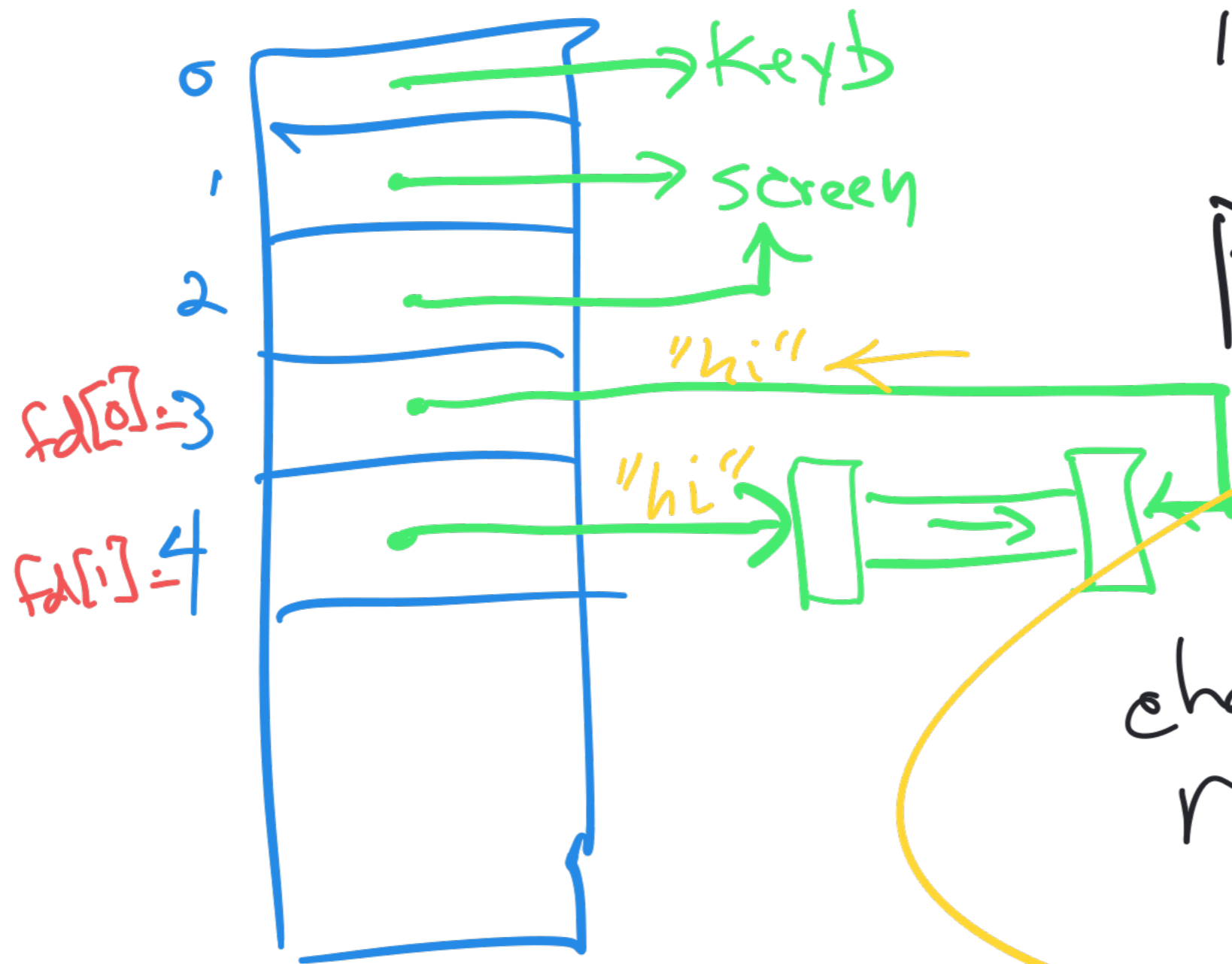




Command `ls -l > out.txt`

`ls -l > out.txt` command

Pipe

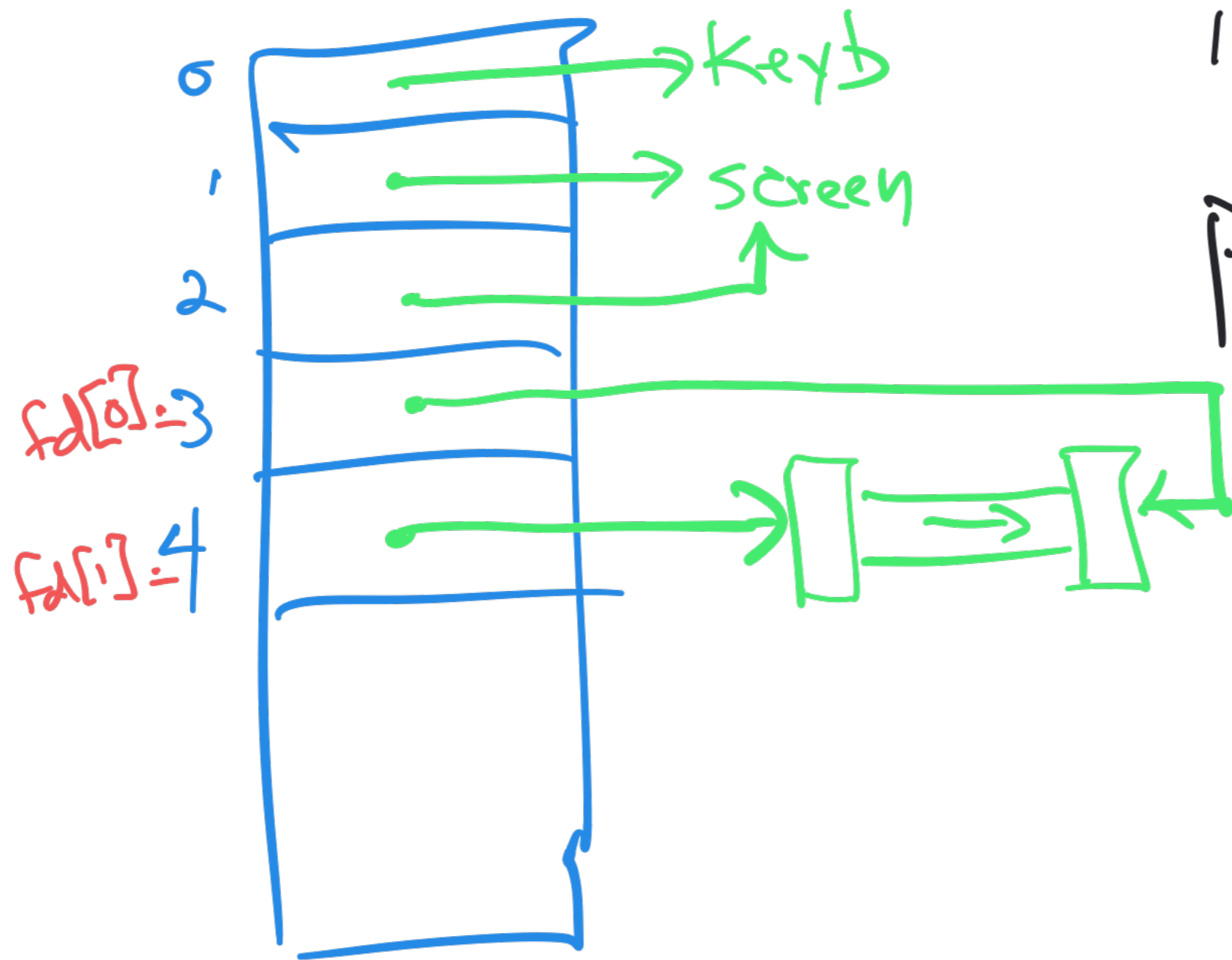


```
int fd[2];  
pipe(fd);
```

```
write(fd[1],  
charbuf[30], "hi", 2);  
read(fd[0], buf, 2);
```

Dumb. Don't do this.

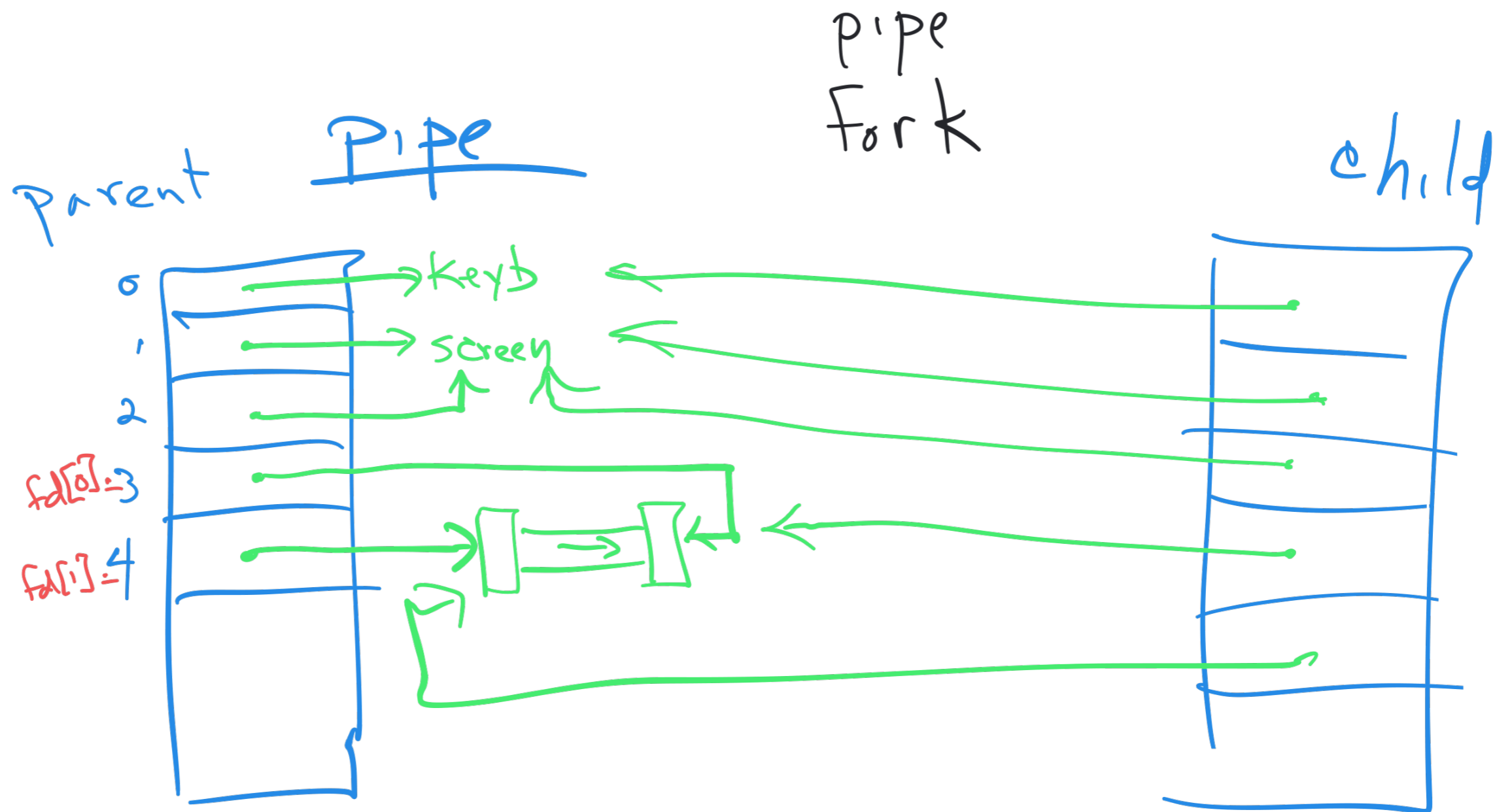
Pipe



```
int fd[2];
```

```
pipe(fd);
```

```
fork()
```

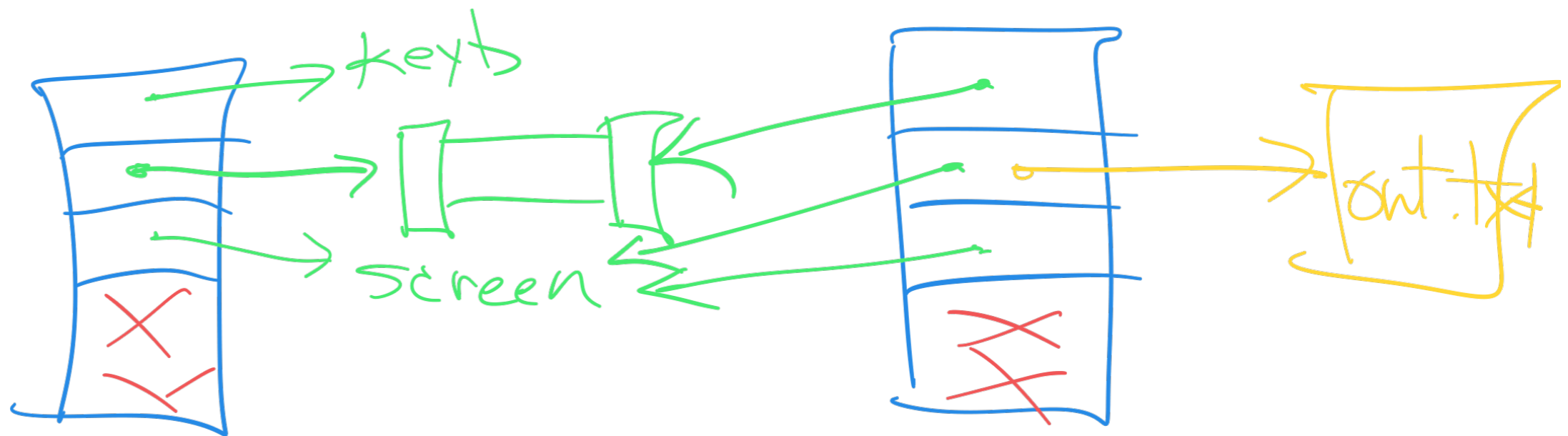
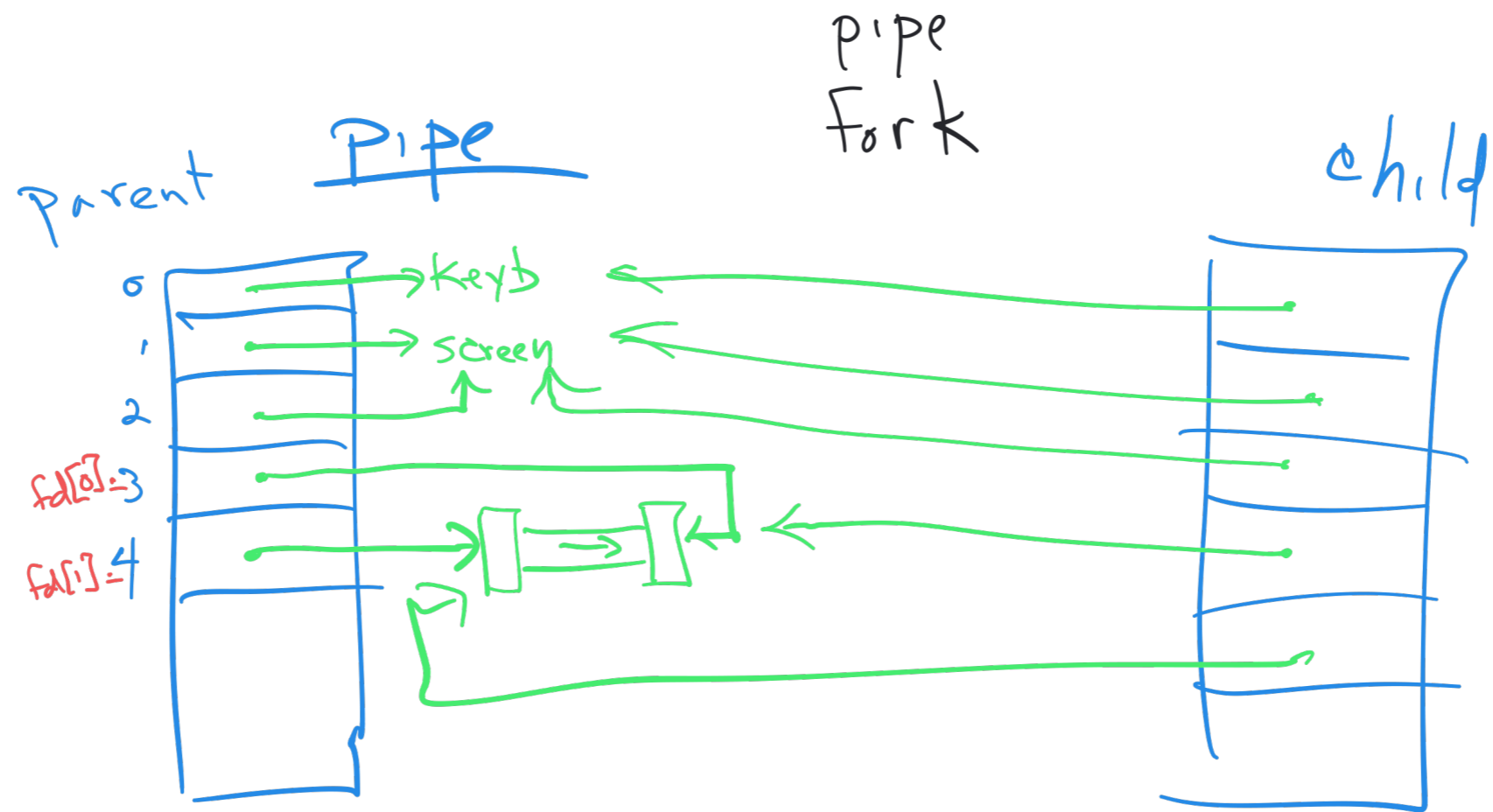


Want parent writes (stdout)
to write-end of pipe
child reads from (stdin)
read-end

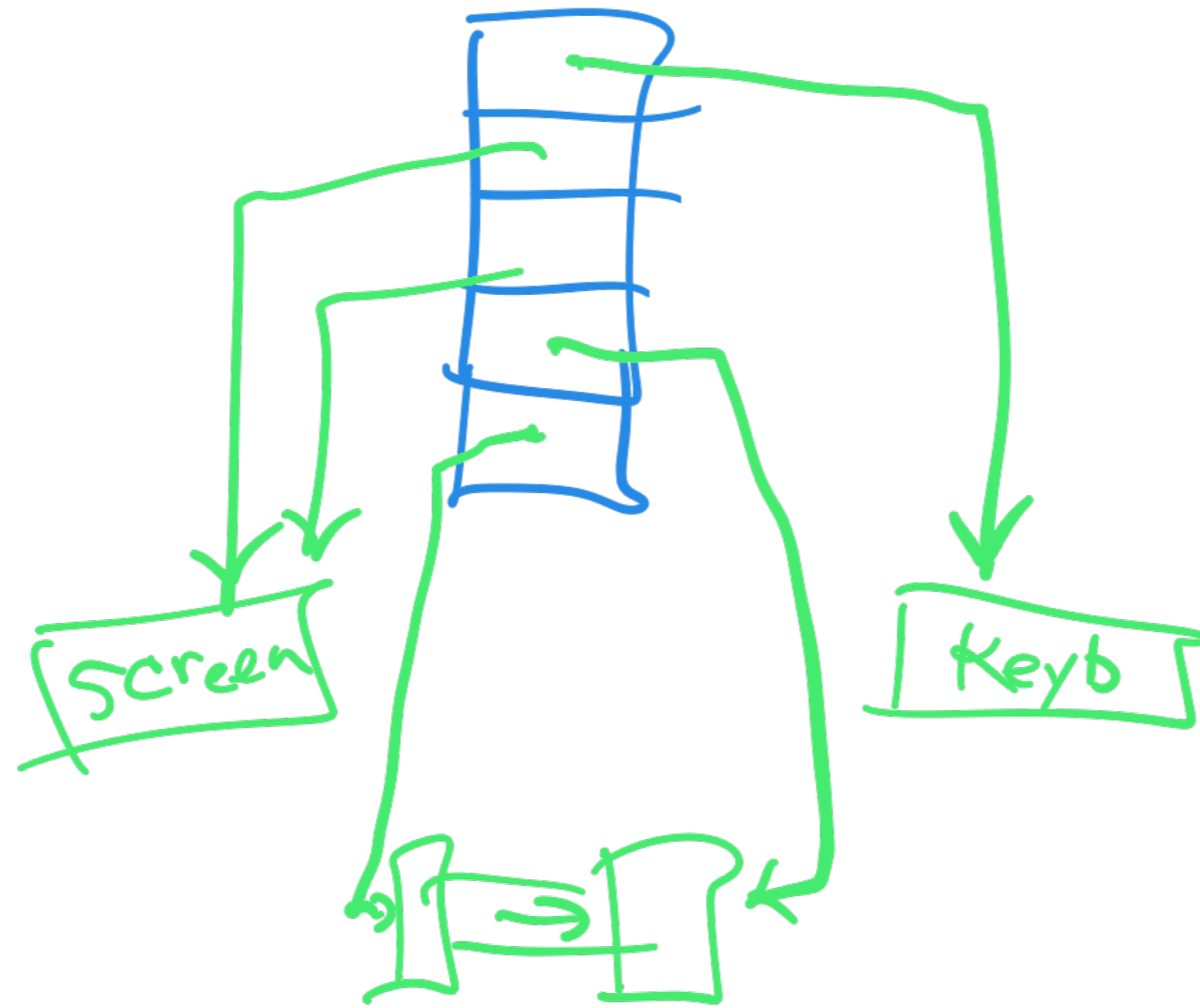
Digression . . .

When a process writes to a pipe & fills it up, the process goes to sleep until some bytes get read from the pipe.

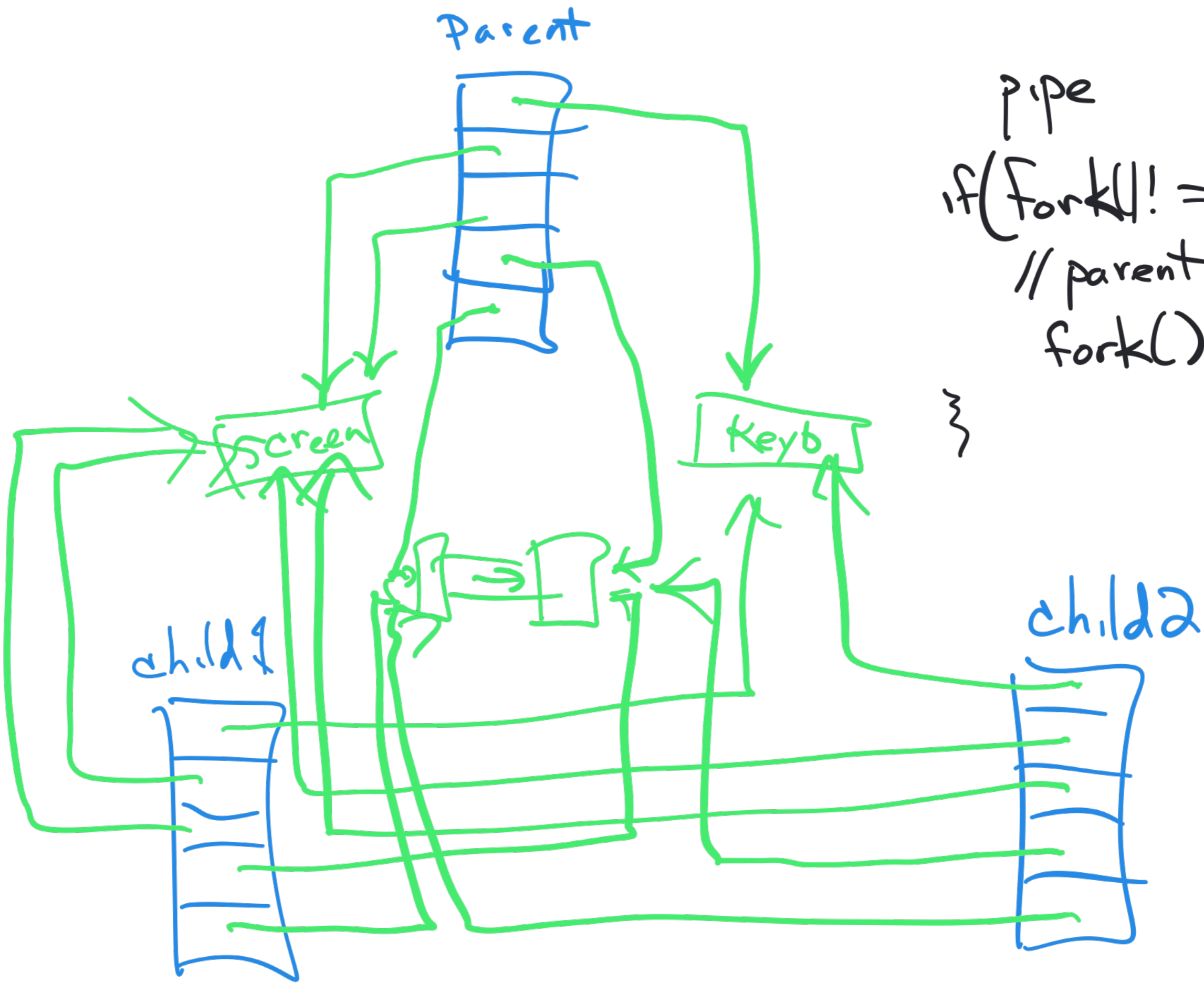
When p reads from empty pipe, ~~sleeps~~ until there's enough data to finish the read.



Parent



pipe

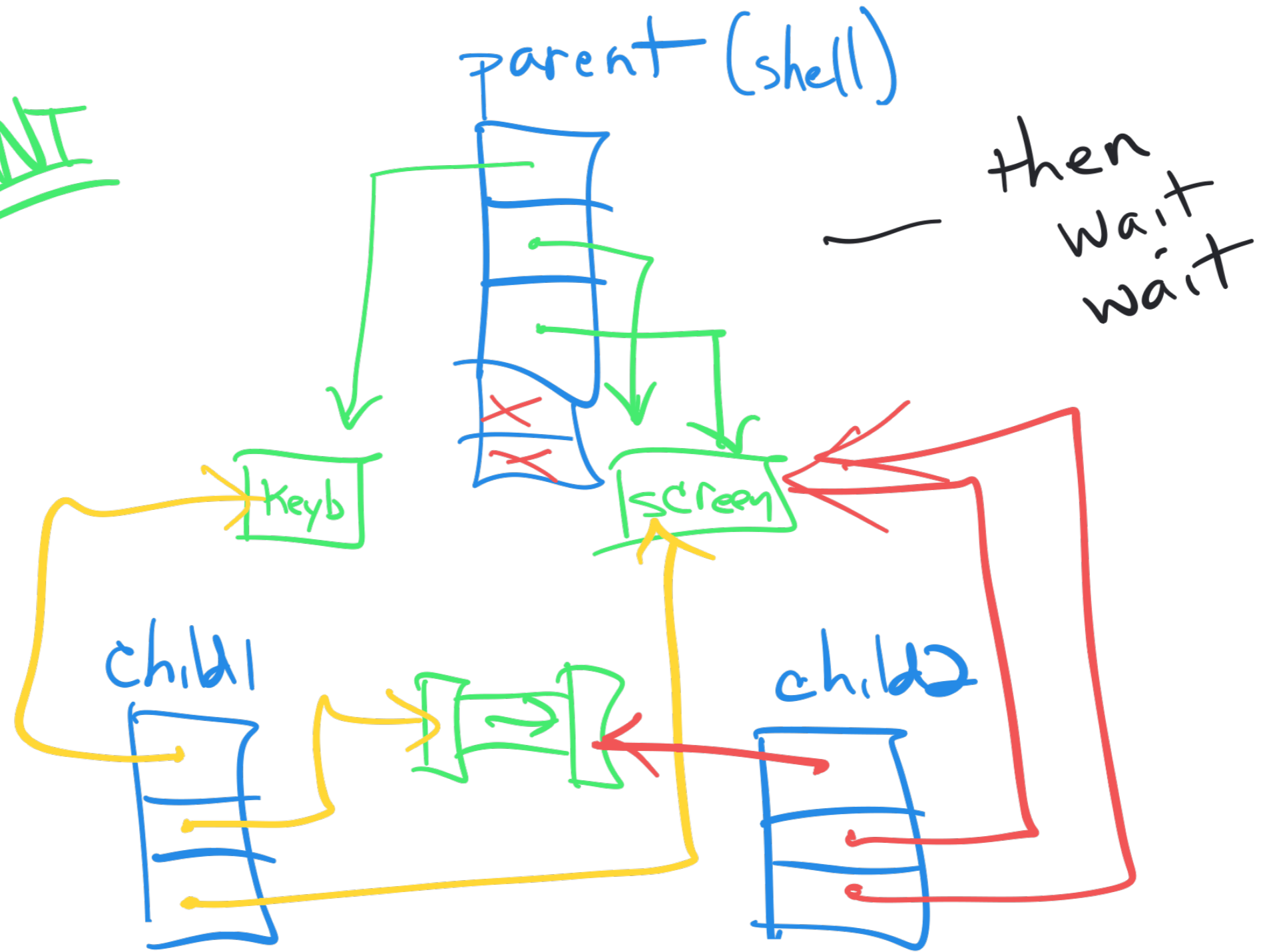


```

pipe
if (fork() != 0) {
  // parent
  fork()
}
~

```

WANT



then... exec in both children