

Consider the following code for implementing a shell.

```
int pid;

void child_is_done(int sig) {
    /* reap child */
    ...
    remove_job(pid);
}

main() {
    ...

    signal(SIGCHLD, child_is_done);

    if ((pid = fork()) == 0) {
        /* child executes */
        ...
        exit(0);
    }

    add_job(pid);

    ...
}
```

There is a possible problem, if the child finishes before the parent gets a chance to run `add_job(pid)`.

How could you solve this problem?