

4. p4.c fork() and I/O redirect

child redirects stdout to file p4.output then runs wc

```
$ cat -n ostep/code/cpu-api/p4.c
```

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <unistd.h>
 4 #include <string.h>
 5 #include <fcntl.h>
 6 #include <assert.h>
 7 #include <sys/wait.h>
 8
 9 int
10 main(int argc, char *argv[])
11 {
12     int rc = fork();
13     if (rc < 0) {
14         // fork failed; exit
15         fprintf(stderr, "fork failed\n");
16         exit(1);
17     } else if (rc == 0) {
18         // child: redirect standard output to a file
19         close(STDOUT_FILENO);
20         open("./p4.output", O_CREAT|O_WRONLY|O_TRUNC, S_IRWXU);
21
22         // now exec "wc"...
23         char *myargs[3];
24         myargs[0] = strdup("wc"); // program: "wc" (word count)
25         myargs[1] = strdup("p4.c"); // argument: file to count
26         myargs[2] = NULL; // marks end of array
27         execvp(myargs[0], myargs); // runs word count
28     } else {
29         // parent goes down this path (original process)
30         int wc = wait(NULL);
31         assert(wc >= 0);
32     }
33     return 0;
34 }
```

```
$ ./p4
```

```
$ cat p4.output
```

```
34 114 884 p4.c
```

```
$
```