

3. p3.c fork() and execvp() example

child runs wc

```
$ cat -n ostep/code/cpu-api/p3.c
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <sys/wait.h>
6
7  int
8  main(int argc, char *argv[])
9  {
10     printf("hello world (pid:%d)\n", (int) getpid());
11     int rc = fork();
12     if (rc < 0) {
13         // fork failed; exit
14         fprintf(stderr, "fork failed\n");
15         exit(1);
16     } else if (rc == 0) {
17         // child (new process)
18         printf("hello, I am child (pid:%d)\n", (int) getpid());
19         char *myargs[3];
20         myargs[0] = strdup("wc"); // program: "wc" (word count)
21         myargs[1] = strdup("p3.c"); // argument: file to count
22         myargs[2] = NULL; // marks end of array
23         execvp(myargs[0], myargs); // runs word count
24         printf("this shouldn't print out");
25     } else {
26         // parent goes down this path (original process)
27         int wc = wait(NULL);
28         printf("hello, I am parent of %d (wc:%d) (pid:%d)\n",
29                rc, wc, (int) getpid());
30     }
31     return 0;
32 }
```

```
$ ./p3
```

```
hello world (pid:22206)
```

```
hello, I am child (pid:22207)
```

```
32 123 966 p3.c
```

```
hello, I am parent of 22207 (wc:22207) (pid:22206)
```

```
$
```