

1. p1.c fork() example

after fork(), parent or child could be scheduled next

```
$ make
gcc -o p1 p1.c -Wall
gcc -o p2 p2.c -Wall
gcc -o p3 p3.c -Wall
gcc -o p4 p4.c -Wall
$ cat -n ostep/code/cpu-api/p1.c
 1  #include <stdio.h>
 2  #include <stdlib.h>
 3  #include <unistd.h>
 4
 5  int
 6  main(int argc, char *argv[])
 7  {
 8      printf("hello world (pid:%d)\n", (int) getpid());
 9      int rc = fork();
10      if (rc < 0) {
11          // fork failed; exit
12          fprintf(stderr, "fork failed\n");
13          exit(1);
14      } else if (rc == 0) {
15          // child (new process)
16          printf("hello, I am child (pid:%d)\n", (int) getpid());
17      } else {
18          // parent goes down this path (original process)
19          printf("hello, I am parent of %d (pid:%d)\n",
20                rc, (int) getpid());
21      }
22      return 0;
23  }
$ ./p1
hello world (pid:21918)
hello, I am parent of 21919 (pid:21918)
hello, I am child (pid:21919)
$ ./p1
hello world (pid:21920)
hello, I am child (pid:21921)
hello, I am parent of 21921 (pid:21920)
$
```