

3. mem.c example

```
$ cat -n ostep/code/intro/mem.c
 1 #include <unistd.h>
 2 #include <stdio.h>
 3 #include <stdlib.h>
 4 #include "common.h"
 5
 6 int main(int argc, char *argv[]) {
 7     if (argc != 2) {
 8         fprintf(stderr, "usage: mem <value>\n");
 9         exit(1);
10     }
11     int *p;
12     p = malloc(sizeof(int));
13     assert(p != NULL);
14     printf("(%d) addr pointed to by p: %p\n", (int) getpid(), p);
15     *p = atoi(argv[1]); // assign value to addr stored in p
16     while (1) {
17         Spin(1);
18         *p = *p + 1;
19         printf("(%d) value of p: %d\n", getpid(), *p);
20     }
21     return 0;
22 }
$ ./mem 10
(3629) addr pointed to by p: 0x55917a0fb260
(3629) value of p: 11
(3629) value of p: 12
(3629) value of p: 13
^C
$
```

```
$ ./mem 10 & ./mem 100 &
[1] 3645
[2] 3646
$ (3645) addr pointed to by p: 0x55ae4b29a260
(3646) addr pointed to by p: 0x55a15c730260
(3645) value of p: 11
(3646) value of p: 101
(3645) value of p: 12
(3646) value of p: 102
...
$ setarch $(uname --machine) --addr-no-randomize /bin
$ ./mem 10 & ./mem 100 &
[1] 3680
[2] 3681
$ (3680) addr pointed to by p: 0x555555756260
(3681) addr pointed to by p: 0x555555756260
(3680) value of p: 11
(3681) value of p: 101
(3680) value of p: 12
(3681) value of p: 102
...
```