Assignment 2/Test cases

Yuqi Zhou, Illinois Institute of Technology 10/05/2018

Test 1 for traps. Test for system call countTraps().

```
1  #include "types.h"
2  #include "stat.h"
3  #include "user.h"
4
5  int
6  main(void)
7  {
8   countTraps();
9   exit();
10 }
```

result:

Total amount of traps : 3

Total amount of syscall : 3

SYS_exec : 1

SYS_sbrk : 1

SYS_countTraps : 1

Test 2 for traps. Test for system calls fork(), wait(), exit().

```
1  #include "types.h"
2  #include "stat.h"
3  #include "user.h"
4
5  void
6  forktest(void)
7  {
8    int pid;
9    pid = fork();
10   if(pid > 0){
11     printf(1,"");
12     pid = wait();
13     printf(1,"");
14   }else if(pid == 0){
```

```
15      printf(1,"");
16      exit();
17    } else {
18      printf(1,"error");
19    }
20  }
21  int
22  main(void)
23  {
24   forktest();
25   countTraps();
26   exit();
27  }
```

---

result:

Total amount of traps : 6

Total amount of syscall : 6

SYS_fork : 1

SYS_exit : 1

SYS_wait : 1

SYS_exec : 1

SYS_sbrk : 1

SYS_countTraps : 1

Test 3 for traps. Test for system calls read(), open(), write(), close().

```
1   #include "types.h"
2   #include "user.h"
3   #include "fcntl.h"
4
5   #define N 100
6
7   struct test {
8       char name;
9       int number;
10  };
11
12  void
13  save(void)
14  {
15      int fd;
16      struct test t;
17      t.name = 's';
```

```
18      t.number = 1;
19
20      fd = open("backup", O_CREATE | O_RDWR);
21      if(fd >= 0) {
22          //printf(1, "ok: create backup file succeed\n");
23      } else {
24          //printf(1, "error: create backup file failed\n");
25          exit();
26      }
27
28      int size = sizeof(t);
29      if(write(fd, &t, size) != size){
30          //printf(1, "error: write to backup file failed\n");
31          exit();
32      }
33      //printf(1, "write ok\n");
34      close(fd);
35  }
36
37  void
38  load(void)
39  {
40      int fd;
41      struct test t;
42
43      fd = open("backup", O_RDONLY);
44      if(fd >= 0) {
45          //printf(1, "ok: read backup file succeed\n");
46      } else {
47          //printf(1, "error: read backup file failed\n");
48          exit();
49      }
50
51      int size = sizeof(t);
52      if(read(fd, &t, size) != size){
53          //printf(1, "error: read from backup file failed\n");
54          exit();
55      }
56      //printf(1, "file contents name %c and number %d", t.name, t.number);
57      //printf(1, "read ok\n");
58      close(fd);
59  }
60
61  int
62  main(void)
63  {
64      save();
```

```
65      load();
66      countTraps();
67      exit();
68  }
```

result:

Total amount of traps : 9

Total amount of syscall : 9

SYS_read : 1

SYS_exec : 1

SYS_sbrk : 1

SYS_open : 2

SYS_write : 1

SYS_close : 2

SYS_countTraps : 1

Test 4 for traps. Test for system calls fork(), exit(), wait() with for loops.

```
1   #include "types.h"
2   #include "stat.h"
3   #include "user.h"
4
5   #define N  1000
6
7   void
8   forktest(void)
9   {
10    int n, pid;
11
12    //printf(1, "fork test\n");
13
14    for(n=0; n<N; n++){
15      pid = fork();
16      if(pid < 0)
17        break;
18      if(pid == 0)
19        exit();
20    }
21
22    if(n == N){
23      printf(1, "fork claimed to work N times!\n", N);
24      exit();
```

```
25    }
26
27    for(; n > 0; n--){
28      if(wait() < 0){
29        //printf(1, "wait stopped early\n");
30        exit();
31      }
32    }
33
34    if(wait() != -1){
35      //printf(1, "wait got too many\n");
36      exit();
37    }
38
39    //printf(1, "fork test OK\n");
40  }
41
42  int
43  main(void)
44  {
45    forktest();
46    countTraps();
47    exit();
48  }
```

---

result:

Total amount of traps : 188

Total amount of syscall : 188

SYS_fork : 62

SYS_exit : 61

SYS_wait : 62

SYS_exec : 1

SYS_sbrk : 1

SYS_countTraps : 1