Assignment 2/Describe each file changed

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

syscall.h

```
1 22a23
2 > #define SYS_countTraps 22
```

defs.h

```
1 123d122
2 < int             countTraps(void);
```

user.h

```
1 25a26
2 > int countTraps(void);
```

sysproc.c

```
1 91a92,97
2 >
3 > int
4 > sys_countTraps(void)
5 > {
6 >   return countTraps();
7 > }
```

usys.S

```
31a32
> SYSCALL(countTraps)
```

syscall.c

```
9a10,11
> //int countSyscall[22] = {0};
> //int countOtherTraps[19] = {0};
105a108
> extern int sys_countTraps(void);
128a132
> [SYS_countTraps]  sys_countTraps
```

proc.c

```
17,73d16
< void assignarray (struct proc *p) {
<   int i = 0;
<   for ( i = 0;i < 22; i++ ) {
<       p->countSyscall[i] = 0;
<    }
<   for( i = 0; i< 19; i++) {
<      p->countOtherTraps[i] = 0;
<    }
<  }
<
< static char *syscallname[] = {
<   [SYS_fork]    "SYS_fork",
<   [SYS_exit]    "SYS_exit",
<   [SYS_wait]    "SYS_wait ",
<   [SYS_pipe]    "SYS_pipe",
<   [SYS_read]    "SYS_read",
<   [SYS_kill]    "SYS_kill",
<   [SYS_exec]    "SYS_exec",
<   [SYS_fstat]   "SYS_fstat",
<   [SYS_chdir]   "SYS_chdir ",
<   [SYS_dup]     "SYS_dup",
<   [SYS_getpid]  "SYS_getpid",
<   [SYS_sbrk]    "SYS_sbrk",
<   [SYS_sleep]   "SYS_sleep",
```

```
26  <   [SYS_uptime]  "SYS_uptime",
27  <   [SYS_open]    "SYS_open",
28  <   [SYS_write]   "SYS_write ",
29  <   [SYS_mknod]   "SYS_mknod",
30  <   [SYS_unlink]  "SYS_unlink",
31  <   [SYS_link]    "SYS_link",
32  <   [SYS_mkdir]   "SYS_mkdir",
33  <   [SYS_close]   "SYS_close",
34  <   [SYS_countTraps]    "SYS_countTraps"
35  <   };
36  <
37  < static char *othertraps[] = {
38  <   [T_DIVIDE]   "T_DIVIDE",
39  <   [T_DEBUG]    "T_DEBUG",
40  <   [T_NMI]      "T_NMI ",
41  <   [T_BRKPT]    "T_BRKPT",
42  <   [T_OFLOW]    "T_OFLOW",
43  <   [T_BOUND]    "T_BOUND",
44  <   [T_ILLOP]    "T_ILLOP",
45  <   [T_DEVICE]   "T_DEVICE",
46  <   [T_DBLFLT]   "T_DBLFLT ",
47  <   [T_TSS-1]    "T_TSS",
48  <   [T_SEGNP-1]  "T_SEGNP",
49  <   [T_STACK-1]  "T_STACK",
50  <   [T_GPFLT-1]  "T_GPFLT",
51  <   [T_PGFLT-1]  "T_PGFLT",
52  <   [T_FPERR-2]  "T_FPERR",
53  <   [T_ALIGN-2]  "T_ALIGN",
54  <   [T_MCHK-2]     "T_MCHK",
55  <   [T_SIMDERR-2]  "T_SIMDERR",
56  <   [18]           "T_DEFAULT"
57  <   };
58  <
59  172c115
60  <           assignarray(p);
61  ---
62  >
63  262,263d204
64  <           //*np->countOtherTraps = *curproc->countOtherTraps;
65  <           //*np->countSyscall = *curproc->countSyscall;
66  295d235
67  <           int i;
68  299c239
69  <
70  ---
71  >
72  314c254
```

```
<
---
>
317,322c257
<          for( i = 0; i< 22; i++) {
<              curproc->parent->countSyscall[i] = curproc->parent->countSyscall[i] + curproc->countSyscall[i];
<              if ( i < 19 )
<                 curproc->parent->countOtherTraps[i] = curproc->parent->countOtherTraps[i] + curproc->countOtherTraps[i];
<             }
<
---
>
603,631d537
<
< int
< countTraps(void) {
<   int i = 0;
<   int totalSyscall = 0;
<   int totalTraps = 0;
<   for (i = 0; i < 22; i++) {
<      totalSyscall = totalSyscall + myproc()->countSyscall[i];
<      }
<   totalTraps = totalSyscall;
<   for (i =0; i < 19; i++) {
<      totalTraps = totalTraps + myproc()->countOtherTraps[i];
<   }
<   cprintf("Total amount of traps : %d\n", totalTraps);
<   cprintf("Total amount of syscall : %d\n", totalSyscall);
<   for (i = 0; i < 22; i++) {
<      if (myproc()->countSyscall[i] != 0){
<         cprintf("%s : %d\n", syscallname[i+1], myproc()->countSyscall[i]);
<         }
<    }
<   for (i = 0; i < 19; i++) {
<      if (myproc()->countOtherTraps[i] != 0) {
<         cprintf("%s : %d\n", othertraps[i], myproc()->countOtherTraps[i]);
<       }
<    }
<
<   return 22;
< }
<
```