

Operating Systems

Lab. Class

Week 3

Project Plan

- 5 projects
 - Install xv6
 - **System call**
 - Scheduling
 - Virtual memory
 - Concurrency
 - File system
- Individual projects

System Call Revisited

- sysproc.c

```
29 int
30 sys_kill(void)
31 {
32     int pid;
33
34     if (argint(0, &pid) < 0)
35         return -1;
36     return kill(pid);
37 }
```

- defs.h

```
106 // proc.c
107 struct proc*   copyproc(struct proc*);
108 void           exit(void);
109 int            fork(void);
110 int            growproc(int);
111 int            kill(int);
112 void           pinit(void);
```

- proc.c

```
411 int
412 kill(int pid)
413 {
414     struct proc *p;
415
416     acquire(&ptable.lock);
417     for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
418         if(p->pid == pid){
419             p->killed = 1;
```

Test with User Program

- Example: kill system call
- kill.c

```
1 #include "types.h"
2 #include "stat.h"
3 #include "user.h"
4
5 int
6 main(int argc, char **argv)
7 {
8     int i;
9
10    if(argc < 2){
11        printf(2, "usage: kill pid...\n");
12        exit();
13    }
14    for(i=1; i<argc; i++)
15        kill(atoi(argv[i]));
16    exit();
17 }
```

Test with User Program (Cont'd)

- Makefile

```
159 UPROGS=\
160     _cat\
161     _echo\
162     _forktest\
163     _grep\
164     _init\
165     _kill\
166     _ln\
167     _ls\
168     _mkdir\
169     _rm\
170     _sh\
171     _stressfs\
172     _usertests\
173     _wc\
174     _zombie\
```

- xv6

```
$ ls
.          1 1 512
..         1 1 512
README    2 2 1973
cat        2 3 14000
echo       2 4 12961
forktest   2 5 8473
grep       2 6 15924
init       2 7 13862
kill       2 8 13093
ln         2 9 12995
ls         2 10 15859
mkdir      2 11 13126
rm         2 12 13103
sh         2 13 25923
stressfs   2 14 14081
usertests  2 15 68544
wc         2 16 14582
zombie     2 17 12727
console    3 18 0
$
```

Project Template Code

- Download from <http://csl.skku.edu/SWE3004S16/Projects>
- Modifications
 - halt system call
 - Halt xv6 program
 - make tarball
 - Compress your source codes into one .tar.gz file for submission
 - You should enter your ID & project no. on Makefile

```
236 # SKKU operating system
237 PROJECTNUM=1
238 STUDENTID=2011311671
239
240 # DO NOT EDIT
241 tarball:
242     make clean
243     tar cvzf ../xv6-project-$(PROJECTNUM)-$(STUDENTID).tar.gz .
```

Project Submission Procedure

- <http://sys.skku.edu>
- Register an account
 - You must type your real name & student ID
 - Other parts are free
- Next steps will be announced

- Since 2nd submission, -5% penalty of the project score
- Every one day delay, -25% penalty of the project score
 - You can use up to 5 *slip* days

Project Grading Procedure

- By executing user program
 - Init process executes user programs
- Project 1 test cases

```
=== TEST START ===  
case 1. get nice value of init process: OK  
case 2. get nice value of non-existing process: OK  
case 3. set nice value of current process: OK  
case 4. set nice value of non-existing process: OK  
case 5. set wrong nice value of current process: OK  
case 6. get nice value of forked process: OK  
=== TEST END ===  
total score: 6/6
```