(All changes made to the xv6 directory, screenshots at bottom.
Apologies if its long, not sure how you guys would prefer it. Also, please attach any suggestions for a future report so that I can know what to improve on.)

So the first thing I did, I added a variable (int exitStatus) to the process control block(PCB). This will allow me to record an exit status for each terminated process.

Next, I changed the exit system call declaration to void exit(int status). This meant that I would have to change the definition of all the user programs have this signature.

- **User.h**
- **Defs.h**
- **Sysproc.c (system calls related to processes)**
- **Proc.c (sheduling and processes)**

The same procedure applied to the wait system call.

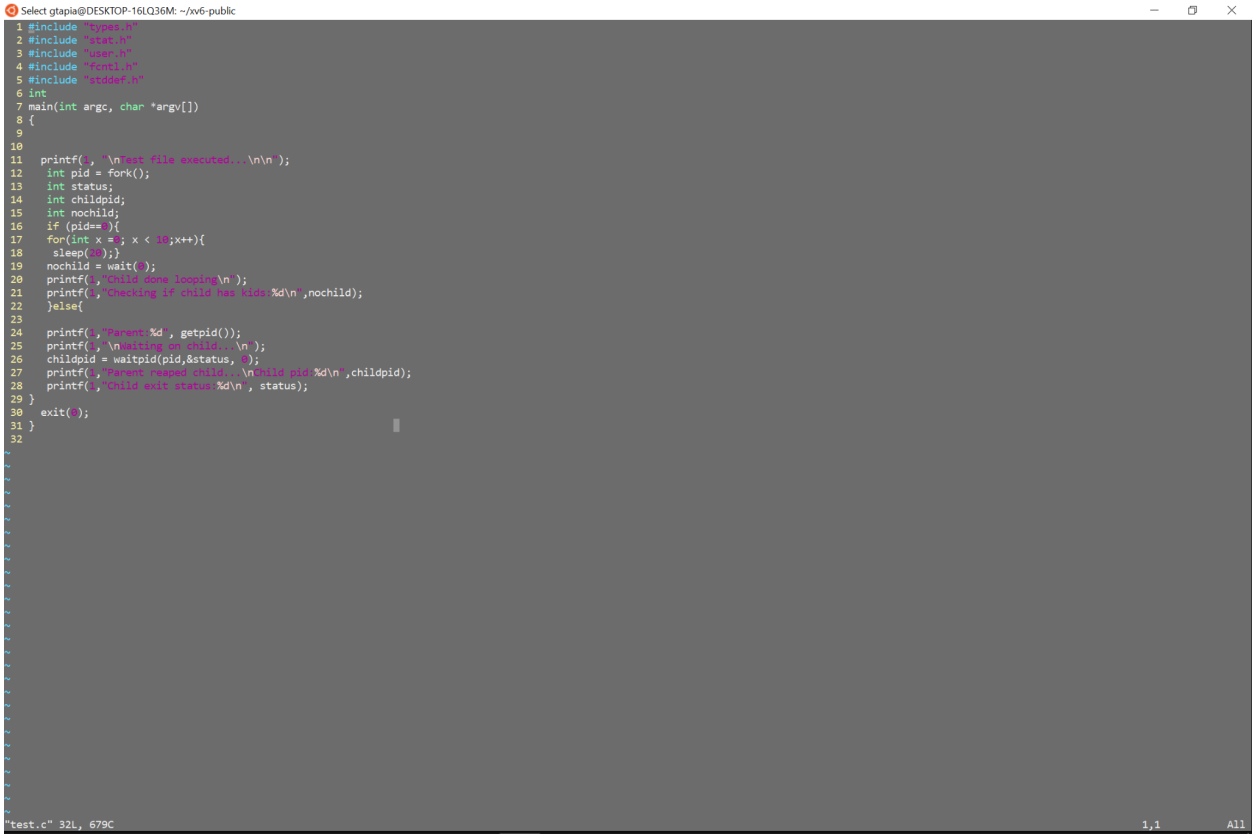
Now that these changes were made, I had to go to all the user space programs, update the wait, and exit calls to include a 0 as their parameter. Doing this allowed me to avoid compilation errors and to satisfy the definition requirements.

For the waitpid system call, I had to register a number with the function. As well as update a bunch files, and implement the body to work similar to wait with some differences. The files modified included:

- **Defs.h (added the declaration for system call)**
- **Syscall.c (adding sys call to sys call table so the kernel can know what sys call it wants. Also added prototype to function.)**
- **Sysproc.c (finding actual system call and implementing it)**
- **User.h (added definition for waitpid so it can be called at terminal)**
- **Syscall.h (associating system call with number, position of system call vector)**
- **Usys.s(here I macor defined waitpid at userspace)**

Finally, for the user program, I had to add it to the makefile so that the program can be built.

(below, I will go over the added functionalities based on the code. I will skip the redundant ones, or simple ones explained above already.)



My program essentially creates a child process by using fork() system call. The child process goes into the if statement and loops for a long time while the parent process goes into the else statement and waits until the child with the specified pid finishes executing. The child process, after the loop checks to see if it has any children by calling wait(0), since it doesn't, it returns negative one, then exits. Parent, then continues once it reaps the child process with the pid provided

gtapia@DESKTOP-16LO36M: ~/xv6-public

```
SeaBIOS (version 1.13.0-1ubuntu1.1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8CA10+1FECCA10 CA00



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ test

Test file executed...

Parent:3
Waiting on child...
```

Select gtapia@DESKTOP-16LO36M: ~/xv6-public

```
SeaBIOS (version 1.13.0-1ubuntu1.1)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8CA10+1FECCA10 CA00



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ test

Test file executed...

Parent:3
Waiting on child...
Child done looping
Checking if child has kids:-1
Parent reaped child...
Child pid:4
Child exit status:1
$
```

```
Only in xv6-public: .gdbinit
Binary files xv6-public/.git/index and xv6-original/.git/index differ
diff -r xv6-public/.git/logs/HEAD xv6-original/.git/logs/HEAD
1c1
< 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1644188560 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
---
> 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1645933203 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
diff -r xv6-public/.git/logs/refs/heads/master xv6-original/.git/logs/refs/heads/master
1c1
< 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1644188560 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
---
> 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1645933203 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
diff -r xv6-public/.git/logs/refs/remotes/origin/HEAD
xv6-original/.git/logs/refs/remotes/origin/HEAD
1c1
< 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1644188560 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
---
> 0000000000000000000000000000000000000000
eeb7b415dbcb12cc362d0783e41c3d1f44066b17 gtapia
<gtapia@DESKTOP-16LQ36M.localdomain> 1645933203 -0600   clone: from
https://github.com/mit-pdos/xv6-public.git
Only in xv6-public/.git/objects/pack: pack-52923efd63d0a59edddfcbe1221e42831049b666.idx
Only in xv6-public/.git/objects/pack: pack-52923efd63d0a59edddfcbe1221e42831049b666.pack
Only in xv6-original/.git/objects/pack: pack-cecc46d53b2e7df5e8d8ef7707bfd93aef05a975.idx
```

```
Only in xv6-original/.git/objects/pack: pack-cecc46d53b2e7df5e8d8ef7707bfd93aef05a975.pack
diff -r xv6-public/Makefile xv6-original/Makefile
30d29
<
185d183
<         _test\
224c222
< QEMUOPTS = -drive file=fs.img,index=1,media=disk,format=raw -drive
file=xv6.img,index=0,media=disk,format=raw -smp $(CPUS) -m 512 $(QEMUEXTRA) -display
none
---
> QEMUOPTS = -drive file=fs.img,index=1,media=disk,format=raw -drive
file=xv6.img,index=0,media=disk,format=raw -smp $(CPUS) -m 512 $(QEMUEXTRA)
254c252
<         ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c test.c\
---
>         ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c\
Only in xv6-public: _cat
Only in xv6-public: _echo
Only in xv6-public: _forktest
Only in xv6-public: _grep
Only in xv6-public: _init
Only in xv6-public: _kill
Only in xv6-public: _ln
Only in xv6-public: _ls
Only in xv6-public: _mkdir
Only in xv6-public: _rm
Only in xv6-public: _sh
Only in xv6-public: _stressfs
Only in xv6-public: _test
Only in xv6-public: _usertests
Only in xv6-public: _wc
Only in xv6-public: _zombie
Only in xv6-public: bio.d
Only in xv6-public: bio.o
Only in xv6-public: bootasm.d
Only in xv6-public: bootasm.o
Only in xv6-public: bootblock
Only in xv6-public: bootblock.asm
Only in xv6-public: bootblock.o
Only in xv6-public: bootblockother.o
Only in xv6-public: bootmain.d
Only in xv6-public: bootmain.o
```

```
Only in xv6-public: cat.asm
diff -r xv6-public/cat.c xv6-original/cat.c
11c11
<
---
>
15c15
<       exit(0);
---
>       exit();
20c20
<     exit(0);
---
>     exit();
31c31
<     exit(0);
---
>     exit();
37c37
<       exit(0);
---
>       exit();
42c42
<   exit(0);
---
>   exit();
Only in xv6-public: cat.d
Only in xv6-public: cat.o
Only in xv6-public: cat.sym
Only in xv6-public: console.d
Only in xv6-public: console.o
diff -r xv6-public/defs.h xv6-original/defs.h
107c107
< void            exit(int);
---
> void            exit(void);
120c120
< int             wait(int*);
---
> int             wait(void);
123c123
< int waitpid(int, int*, int);
---
```

```
>
Only in xv6-public: echo.asm
diff -r xv6-public/echo.c xv6-original/echo.c
12c12
<   exit(0);
---
>   exit();
Only in xv6-public: echo.d
Only in xv6-public: echo.o
Only in xv6-public: echo.sym
Only in xv6-public: entry.o
Only in xv6-public: entryother
Only in xv6-public: entryother.asm
Only in xv6-public: entryother.d
Only in xv6-public: entryother.o
Only in xv6-public: exec.d
Only in xv6-public: exec.o
Only in xv6-public: file.d
Only in xv6-public: file.o
Only in xv6-public: forktest.asm
diff -r xv6-public/forktest.c xv6-original/forktest.c
28c28
<       exit(0);
---
>       exit();
33c33
<     exit(0);
---
>     exit();
37c37
<     if(wait(0) < 0){
---
>     if(wait() < 0){
39c39
<       exit(0);
---
>       exit();
43c43
<   if(wait(0) != -1){
---
>   if(wait() != -1){
45c45
<     exit(0);
```

```
---
>     exit();
55c55
<   exit(0);
---
>   exit();
Only in xv6-public: forktest.d
Only in xv6-public: forktest.o
Only in xv6-public: fs.d
Only in xv6-public: fs.img
Only in xv6-public: fs.o
Only in xv6-public: grep.asm
diff -r xv6-public/grep.c xv6-original/grep.c
46c46
<     exit(0);
---
>     exit();
52c52
<     exit(0);
---
>     exit();
58c58
<       exit(0);
---
>       exit();
63c63
<   exit(0);
---
>   exit();
Only in xv6-public: grep.d
Only in xv6-public: grep.o
Only in xv6-public: grep.sym
Only in xv6-public: ide.d
Only in xv6-public: ide.o
Only in xv6-public: init.asm
diff -r xv6-public/init.c xv6-original/init.c
27c27
<       exit(0);
---
>       exit();
32c32
<       exit(0);
---
```

```
>       exit();
34c34
<     while((wpid=wait(0)) >= 0 && wpid != pid)
---
>     while((wpid=wait()) >= 0 && wpid != pid)
Only in xv6-public: init.d
Only in xv6-public: init.o
Only in xv6-public: init.sym
Only in xv6-public: initcode
Only in xv6-public: initcode.asm
Only in xv6-public: initcode.d
Only in xv6-public: initcode.o
Only in xv6-public: initcode.out
Only in xv6-public: ioapic.d
Only in xv6-public: ioapic.o
Only in xv6-public: kalloc.d
Only in xv6-public: kalloc.o
Only in xv6-public: kbd.d
Only in xv6-public: kbd.o
Only in xv6-public: kernel
Only in xv6-public: kernel.asm
Only in xv6-public: kernel.sym
Only in xv6-public: kill.asm
diff -r xv6-public/kill.c xv6-original/kill.c
12c12
<     exit(0);
---
>     exit();
16c16
<   exit(0);
---
>   exit();
Only in xv6-public: kill.d
Only in xv6-public: kill.o
Only in xv6-public: kill.sym
Only in xv6-public: lapic.d
Only in xv6-public: lapic.o
Only in xv6-public: ln.asm
diff -r xv6-public/ln.c xv6-original/ln.c
10c10
<     exit(0);
---
>     exit();
```

```
14c14
<   exit(0);
---
>   exit();
Only in xv6-public: ln.d
Only in xv6-public: ln.o
Only in xv6-public: ln.sym
Only in xv6-public: log.d
Only in xv6-public: log.o
Only in xv6-public: ls.asm
diff -r xv6-public/ls.c xv6-original/ls.c
80c80
<     exit(0);
---
>     exit();
84c84
<   exit(0);
---
>   exit();
Only in xv6-public: ls.d
Only in xv6-public: ls.o
Only in xv6-public: ls.sym
Only in xv6-public: main.d
Only in xv6-public: main.o
Only in xv6-public: mkdir.asm
diff -r xv6-public/mkdir.c xv6-original/mkdir.c
12c12
<     exit(0);
---
>     exit();
22c22
<   exit(0);
---
>   exit();
Only in xv6-public: mkdir.d
Only in xv6-public: mkdir.o
Only in xv6-public: mkdir.sym
Only in xv6-public: mkfs
Only in xv6-public: mp.d
Only in xv6-public: mp.o
Only in xv6-public: picirq.d
Only in xv6-public: picirq.o
Only in xv6-public: pipe.d
```

```
Only in xv6-public: pipe.o
Only in xv6-public: printf.d
Only in xv6-public: printf.o
diff -r xv6-public/proc.c xv6-original/proc.c
228c228
< exit(int status)
---
> exit(void)
233,235c233
```
Made the change here so that the current process can have a status associated with it. One being successful
```
<
<   status=1;
<   curproc->exitStatus= status;
---
>
237a236
>
274c273
< wait(int *status)
---
> wait(void)
279c278
<
---
>
300,301d298
```
The if statement is for when the child process referenced from the process table status != 0 then we set its current exit status of the zombie child as 1 for successful termination
```
<       if(status!=0)
<       { *status = p->exitStatus;}

316,351d312
```
Here, I pretty much copied over the body of the wait function since waitpid works similar to it, just with seperate functionalities, including status and options. I added a zombie pid to be recorded and set the status of the zombie process as successful(1), if it terminated successfully.

The current process waits on its children to terminate and returns -1 if no children.
```
< int waitpid (int pid, int* status, int options)
< {
<   struct proc *p;
<   struct proc *curproc = myproc();
```

```
<   int process;
<   int zombiepid;
<
<   acquire(&ptable.lock);
<   for(;;){
<     process = 0;
<     for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
<         if(p->pid != pid)
<         continue;
<       process = 1;
<       if(p->state == ZOMBIE){
<         zombiepid = p->pid;
<         kfree(p->kstack);
<         p->kstack = 0;
<         freevm(p->pgdir);
<         p->state = UNUSED;
<         p->pid = 0;
<         p->parent = 0;
<         p->name[0] = 0;
<         p->killed = 0;
<         release(&ptable.lock);
<          if (status != 0)
<         {  *status = p->exitStatus;}
<          return zombiepid;
<       }
<     }
<
<     // No point waiting if we don't have any children.
<     if(!process || curproc->killed){
<       release(&ptable.lock);
<       return -1;
<     }
353,356d313
<     // Wait for children to exit.  (See wakeup1 call in proc_exit.)
<     sleep(curproc, &ptable.lock);  //DOC: wait-sleep
<   }
< }
359c316
< // Each CPU calls s/.cheduler() after setting itself up.
---
> // Each CPU calls scheduler() after setting itself up.
Only in xv6-public: proc.d
diff -r xv6-public/proc.h xv6-original/proc.h
```

```
52,53d51
<   int ext; // add interupt here
<   int exitStatus;                 // Process exit status
Only in xv6-public: proc.o
Only in xv6-public: rm.asm
diff -r xv6-public/rm.c xv6-original/rm.c
12c12
<     exit(0);
---
>     exit();
22c22
<   exit(0);
---
>   exit();
Only in xv6-public: rm.d
Only in xv6-public: rm.o
Only in xv6-public: rm.sym
Only in xv6-public: sh.asm
diff -r xv6-public/sh.c xv6-original/sh.c
68c68
<     exit(0);
---
>     exit();
77c77
<       exit(0);
---
>       exit();
87c87
<       exit(0);
---
>       exit();
96c96
<     wait(0);
---
>     wait();
120,121c120,121
<     wait(0);
<     wait(0);
---
>     wait();
>     wait();
130c130
<   exit(0);
```

```
---
>   exit();
169c169
<     wait(0);
---
>     wait();
171c171
<   exit(0);
---
>   exit();
178c178
<   exit(0);
---
>   exit();
Only in xv6-public: sh.d
Only in xv6-public: sh.o
Only in xv6-public: sh.sym
Only in xv6-public: sleeplock.d
Only in xv6-public: sleeplock.o
Only in xv6-public: spinlock.d
Only in xv6-public: spinlock.o
Only in xv6-public: stressfs.asm
diff -r xv6-public/stressfs.c xv6-original/stressfs.c
46c46
<   wait(0);
---
>   wait();
48c48
<   exit(0);
---
>   exit();
Only in xv6-public: stressfs.d
Only in xv6-public: stressfs.o
Only in xv6-public: stressfs.sym
Only in xv6-public: string.d
Only in xv6-public: string.o
Only in xv6-public: swtch.o
diff -r xv6-public/syscall.c xv6-original/syscall.c
106d105
< extern int sys_waitpid(void);
130d128
< [SYS_waitpid] sys_waitpid,
Only in xv6-public: syscall.d
```

```
diff -r xv6-public/syscall.h xv6-original/syscall.h
23d22
< #define SYS_waitpid 22
Only in xv6-public: syscall.o
Only in xv6-public: sysfile.d
Only in xv6-public: sysfile.o
diff -r xv6-public/sysproc.c xv6-original/sysproc.c
16,24d15
< void
```
Here, I updated the wrapper functions in this file sysproc.c. The system calls within the kernel have to find the arguments passed in by user space. Because the arguments are typically in registers, the kernel trap handling procedure saves the context of the user registers to the current trap frame. These wrapper functions are here to help retrieve these arguments. The functions include argint, and argptr, from the ones I used. However, there are others.
```
< sys_exit(int status){
<  int i;
<  if(argint(0, &i)<0)
<     exit(0);
<   exit(status);
<  // return 0;  // not reached
< }
<
26,30c17,20
< sys_wait(int *status)
< { char* temp;
<   argptr(0,&temp,4);
<   int* report = (int*)temp;
<   return wait(report);
---
> sys_exit(void)
> {
>   exit();
>   return 0;  // not reached
33,34c23,24
< int
< sys_waitpid(int pid, int *status, int options)
---
> int

> sys_wait(void)
36,44c26
<    char *temp;
<    int *report;
```

```
<    argptr(1,&temp,4);
<    report = (int*)temp;
<    int id;
<    int opt;
<    argint(0,&id);
<    argint(2,&opt);
<    return waitpid(id,report, opt);
---
>   return wait();
50a33
>
Only in xv6-public: sysproc.d
Only in xv6-public: sysproc.o
Only in xv6-public: test.asm
Only in xv6-public: test.c
Only in xv6-public: test.d
Only in xv6-public: test.o
Only in xv6-public: test.sym
diff -r xv6-public/trap.c xv6-original/trap.c
41c41
<       exit(0);
---
>       exit();
45c45
<       exit(0);
---
>       exit();
101c101
<     exit(0);
---
>     exit();
111c111
<     exit(0);
---
>     exit();
Only in xv6-public: trap.d
Only in xv6-public: trap.o
Only in xv6-public: trapasm.o
Only in xv6-public: uart.d
Only in xv6-public: uart.o
Only in xv6-public: ulib.d
Only in xv6-public: ulib.o
Only in xv6-public: umalloc.d
```

```
Only in xv6-public: umalloc.o
diff -r xv6-public/user.h xv6-original/user.h
6,7c6,7
< int exit(int status) __attribute__((noreturn));
< int wait(int *status);
---
> int exit(void) __attribute__((noreturn));
> int wait(void);
26d25
< int waitpid(int, int*, int);
Only in xv6-public: usertests.asm
diff -r xv6-public/usertests.c xv6-original/usertests.c
24c24
<     exit(0);
---
>     exit();
28c28
<     exit(0);
---
>     exit();
32c32
<     exit(0);
---
>     exit();
36c36
<     exit(0);
---
>     exit();
41c41
< // does exit(0) call iput(p->cwd) in a transaction?
---
> // does exit() call iput(p->cwd) in a transaction?
52c52
<     exit(0);
---
>     exit();
57c57
<       exit(0);
---
>       exit();
61c61
<       exit(0);
---
```

```
>       exit();
65c65
<       exit(0);
---
>       exit();
67c67
<     exit(0);
---
>     exit();
69c69
<   wait(0);
---
>   wait();
92c92
<     exit(0);
---
>     exit();
97c97
<     exit(0);
---
>     exit();
103c103
<       exit(0);
---
>       exit();
105c105
<     exit(0);
---
>     exit();
110c110
<     exit(0);
---
>     exit();
112c112
<   wait(0);
---
>   wait();
127c127
<     exit(0);
---
>     exit();
133c133
<     exit(0);
```

```
---
>     exit();
150c150
<     exit(0);
---
>     exit();
155c155
<       exit(0);
---
>       exit();
159c159
<       exit(0);
---
>       exit();
169c169
<     exit(0);
---
>     exit();
176c176
<     exit(0);
---
>     exit();
182c182
<     exit(0);
---
>     exit();
197c197
<     exit(0);
---
>     exit();
204c204
<       exit(0);
---
>       exit();
213c213
<     exit(0);
---
>     exit();
222c222
<         exit(0);
---
>         exit();
227c227
```

```
<       exit(0);
---
>       exit();
232c232
<       exit(0);
---
>       exit();
239c239
<     exit(0);
---
>     exit();
273c273
<     exit(0);
---
>     exit();
278c278
<     exit(0);
---
>     exit();
283c283
<     exit(0);
---
>     exit();
288c288
<     exit(0);
---
>     exit();
299c299
<     exit(0);
---
>     exit();
313c313
<     exit(0);
---
>     exit();
324c324
<         exit(0);
---
>         exit();
327c327
<     exit(0);
---
>     exit();
```

```
346c346
<       exit(0);
---
>       exit();
349c349
<     wait(0);
---
>     wait();
352c352
<     exit(0);
---
>     exit();
397,399c397,399
<   wait(0);
<   wait(0);
<   wait(0);
---
>   wait();
>   wait();
>   wait();
416c416
<       if(wait(0) != pid){
---
>       if(wait() != pid){
421c421
<       exit(0);
---
>       exit();
450c450
<       exit(0);
---
>       exit();
454c454
<     exit(0);
---
>     exit();
456c456
<     wait(0);
---
>     wait();
487c487
<     exit(0);
---
```

```
>     exit();
489c489
<     wait(0);
---
>     wait();
511c511
<     exit(0);
---
>     exit();
533c533
<       exit(0);
---
>       exit();
540c540
<         exit(0);
---
>         exit();
547c547
<           exit(0);
---
>           exit();
550c550
<       exit(0);
---
>       exit();
555c555
<     wait(0);
---
>     wait();
566c566
<           exit(0);
---
>           exit();
574c574
<       exit(0);
---
>       exit();
596c596
<       exit(0);
---
>       exit();
607c607
<           exit(0);
```

```
---
>          exit();
614c614
<            exit(0);
---
>            exit();
618c618
<       exit(0);
---
>       exit();
623c623
<     wait(0);
---
>     wait();
634c634
<         exit(0);
---
>         exit();
637c637
<         exit(0);
---
>         exit();
665c665
<     exit(0);
---
>     exit();
673c673
<     exit(0);
---
>     exit();
677c677
<     exit(0);
---
>     exit();
686c686
<     exit(0);
---
>     exit();
690c690
<     exit(0);
---
>     exit();
694c694
```

```
<     exit(0);
---
>     exit();
714c714
<     exit(0);
---
>     exit();
718c718
<     exit(0);
---
>     exit();
724c724
<     exit(0);
---
>     exit();
730c730
<     exit(0);
---
>     exit();
736c736
<     exit(0);
---
>     exit();
740c740
<     exit(0);
---
>     exit();
746c746
<     exit(0);
---
>     exit();
752c752
<     exit(0);
---
>     exit();
757c757
<     exit(0);
---
>     exit();
790c790
<         exit(0);
---
>         exit();
```

```
795c795
<       exit(0);
---
>       exit();
797c797
<       wait(0);
---
>       wait();
810c810
<         exit(0);
---
>         exit();
814c814
<         exit(0);
---
>         exit();
824c824
<     exit(0);
---
>     exit();
832c832
<       exit(0);
---
>       exit();
847c847
<       exit(0);
---
>       exit();
849c849
<       wait(0);
---
>       wait();
868c868
<     exit(0);
---
>     exit();
884c884
<     wait(0);
---
>     wait();
886c886
<     exit(0);
---
```

```
>     exit();
904c904
<     exit(0);
---
>     exit();
915c915
<       exit(0);
---
>       exit();
927c927
<       exit(0);
---
>       exit();
944c944
<     exit(0);
---
>     exit();
950c950
<     exit(0);
---
>     exit();
957c957
<     exit(0);
---
>     exit();
962c962
<     exit(0);
---
>     exit();
968c968
<     exit(0);
---
>     exit();
976c976
<     exit(0);
---
>     exit();
981c981
<     exit(0);
---
>     exit();
987c987
<     exit(0);
```

```
---
>     exit();
992c992
<     exit(0);
---
>     exit();
996c996
<     exit(0);
---
>     exit();
1001c1001
<     exit(0);
---
>     exit();
1005c1005
<     exit(0);
---
>     exit();
1009c1009
<     exit(0);
---
>     exit();
1013c1013
<     exit(0);
---
>     exit();
1019c1019
<     exit(0);
---
>     exit();
1023c1023
<     exit(0);
---
>     exit();
1029c1029
<     exit(0);
---
>     exit();
1034c1034
<     exit(0);
---
>     exit();
1038c1038
```

```
<     exit(0);
---
>     exit();
1042c1042
<     exit(0);
---
>     exit();
1046c1046
<     exit(0);
---
>     exit();
1050c1050
<     exit(0);
---
>     exit();
1054c1054
<     exit(0);
---
>     exit();
1058c1058
<     exit(0);
---
>     exit();
1062c1062
<     exit(0);
---
>     exit();
1066c1066
<     exit(0);
---
>     exit();
1070c1070
<     exit(0);
---
>     exit();
1074c1074
<     exit(0);
---
>     exit();
1078c1078
<     exit(0);
---
>     exit();
```

```
1082c1082
<     exit(0);
---
>     exit();
1086c1086
<     exit(0);
---
>     exit();
1090c1090
<     exit(0);
---
>     exit();
1095c1095
<     exit(0);
---
>     exit();
1099c1099
<     exit(0);
---
>     exit();
1103c1103
<     exit(0);
---
>     exit();
1107c1107
<     exit(0);
---
>     exit();
1111c1111
<     exit(0);
---
>     exit();
1130c1130
<       exit(0);
---
>       exit();
1137c1137
<         exit(0);
---
>         exit();
1158c1158
<     exit(0);
---
```

```
>     exit();
1164c1164
<       exit(0);
---
>       exit();
1172c1172
<     exit(0);
---
>     exit();
1179c1179
<       exit(0);
---
>       exit();
1185c1185
<       exit(0);
---
>       exit();
1189c1189
<       exit(0);
---
>       exit();
1196c1196
<     exit(0);
---
>     exit();
1213c1213
<     exit(0);
---
>     exit();
1217c1217
<     exit(0);
---
>     exit();
1222c1222
<     exit(0);
---
>     exit();
1228c1228
<     exit(0);
---
>     exit();
1234c1234
<     exit(0);
```

```
---
>     exit();
1238c1238
<     exit(0);
---
>     exit();
1250c1250
<     exit(0);
---
>     exit();
1254c1254
<     exit(0);
---
>     exit();
1258c1258
<     exit(0);
---
>     exit();
1262c1262
<     exit(0);
---
>     exit();
1266c1266
<     exit(0);
---
>     exit();
1270c1270
<     exit(0);
---
>     exit();
1274c1274
<     exit(0);
---
>     exit();
1278c1278
<     exit(0);
---
>     exit();
1293c1293
<     exit(0);
---
>     exit();
1298c1298
```

```
<     exit(0);
---
>     exit();
1303c1303
<     exit(0);
---
>     exit();
1308c1308
<     exit(0);
---
>     exit();
1312c1312
<     exit(0);
---
>     exit();
1316c1316
<     exit(0);
---
>     exit();
1320c1320
<     exit(0);
---
>     exit();
1324c1324
<     exit(0);
---
>     exit();
1330c1330
<     exit(0);
---
>     exit();
1335c1335
<     exit(0);
---
>     exit();
1354c1354
<       exit(0);
---
>       exit();
1358c1358
<       exit(0);
---
>       exit();
```

```
1391c1391
<       exit(0);
---
>       exit();
1396c1396
<     exit(0);
---
>     exit();
1400c1400
<     if(wait(0) < 0){
---
>     if(wait() < 0){
1402c1402
<       exit(0);
---
>       exit();
1406c1406
<   if(wait(0) != -1){
---
>   if(wait() != -1){
1408c1408
<     exit(0);
---
>     exit();
1431c1431
<       exit(0);
---
>       exit();
1439c1439
<     exit(0);
---
>     exit();
1445c1445
<     exit(0);
---
>     exit();
1448,1449c1448,1449
<     exit(0);
<   wait(0);
---
>     exit();
>   wait();
1458c1458
```

```
<     exit(0);
---
>     exit();
1468c1468
<     exit(0);
---
>     exit();
1473c1473
<     exit(0);
---
>     exit();
1481c1481
<     exit(0);
---
>     exit();
1486c1486
<     exit(0);
---
>     exit();
1493c1493
<     exit(0);
---
>     exit();
1502c1502
<       exit(0);
---
>       exit();
1507c1507
<       exit(0);
---
>       exit();
1509c1509
<     wait(0);
---
>     wait();
1516c1516
<     exit(0);
---
>     exit();
1536c1536
<     wait(0);
---
>     wait();
```

```
1540c1540
<     exit(0);
---
>     exit();
1575c1575
<       exit(0);
---
>       exit();
1580c1580
<     wait(0);
---
>     wait();
1585c1585
<       exit(0);
---
>       exit();
1603c1603
<       exit(0);
---
>       exit();
1630c1630
<     exit(0);
---
>     exit();
1633c1633
<     exit(0);
---
>     exit();
1635c1635
<   wait(0);
---
>   wait();
1639c1639
<     exit(0);
---
>     exit();
1718c1718
<     exit(0);
---
>     exit();
1721c1721
<     exit(0);
---
```

```
>     exit();
1723c1723
<   wait(0);
---
>   wait();
1733c1733
<     exit(0);
---
>     exit();
1755c1755
<     exit(0);
---
>     exit();
1802c1802
<   exit(0);
---
>   exit();
Only in xv6-public: usertests.d
Only in xv6-public: usertests.o
Only in xv6-public: usertests.sym
diff -r xv6-public/usys.S xv6-original/usys.S
32d31
< SYSCALL(waitpid)
Only in xv6-public: usys.o
Only in xv6-public: vectors.S
Only in xv6-public: vectors.o
Only in xv6-public: vm.d
Only in xv6-public: vm.o
Only in xv6-public: wc.asm
diff -r xv6-public/wc.c xv6-original/wc.c
30c30
<     exit(0);
---
>     exit();
42c42
<     exit(0);
---
>     exit();
48c48
<       exit(0);
---
>       exit();
53c53
```

```
<   exit(0);
---
>   exit();
Only in xv6-public: wc.d
Only in xv6-public: wc.o
Only in xv6-public: wc.sym
Only in xv6-public: xv6.img
Only in xv6-public: zombie.asm
diff -r xv6-public/zombie.c xv6-original/zombie.c
13c13
<   exit(0);
---
>   exit();
Only in xv6-public: zombie.d
Only in xv6-public: zombie.o
Only in xv6-public: zombie.sym
```