

Reminders

- n Project 1 due tomorrow by 6:00
- n Office hours in 006 today at 4:30
- Start thinking about project groups (3) people) for the rest of the quarter
 - Groups due by Tuesday noon (email anm@cs)
 - n After that we'll pick a group for you
- n Today: project 1 questions



Project 1 – issues

- _n C strings
- n Copy to/from user and counters
- Syscalls: macros; arguments
- _n Execvp, wait
- Other things??



C strings

- n You only need to use:
 - **strncmp**(src,dest,256) compare strings, 0 if equal, not 0 o.w. Do not do str1 == str2!!!
 - strtok:

 - 1st use: tok = strtok(buf, "delimiters"); Subsequent uses: tok = strtok(NULL, "delimiters");
 - **fgets**(buf, 256, stdin) read a line (up to 256 chars) from stdin (or getline)
 - (maybe) **strncpy**(dest, src, 256) copy up to 256 chars from src to dest.
 - (maybe) Allocate memory with malloc, free with free
- n Fine to assume:
 - ⁿ A maximum length for a shell command (say, 256)
 - Maximum number of arguments (say, 256 again)



Passing counters

- Do not printk the statistics in execcounts!!!
- Execcounts should pass count values to the shell
 - The shell then prints out statistics
- n Copying counters to userspace:
 - Shell passes in something to hold data
 - n sys_execcounts fills the data in



Copying data to/from kernel

unsafe to directly access user pointers!

```
long sys_gettimeofday(struct timeval *tv)
   if (tv) {
         struct timeval ktv;
         do_gettimeofday(&ktv);
         if (copy_to_user(tv, &ktv, sizeof(ktv)))
               return -EFAULT;
```

n copy_to/from_user return amount of uncopied

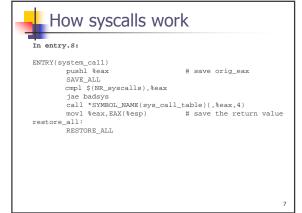
Syscalls

- n Two ways to use one:
 - Linux style:
 - in <asm/unistd.h>:

```
#define __NR_foo 292
static inline _syscall2(int, foo, int, arg1,
                       char *, arg2)
```

- n In userspace, just call foo(4, "test");
- BSD style:

```
n in shell.c:
#define __NR_foo 292
ret = syscall(__nr_foo, arg1, arg2);
```





- n execvp:
 - n You must build an array of strings to pass to it
 - n Make sure the last thing in this array is NULL
 - Make sure the array includes the program name

8



Wait

- n wait(int *status)
 - "man 2 wait" get information about it.
 "man wait" by default goes to the shell reference!
 - What's wrong with this code:

int *status;
wait(status);

Fix??

9



Extern

How do we access global variables defined in one file from another file?

10



Other things

- n Check that every malloc has a matching free
- _n Check for all errors
 - E.g. malloc returns NULL
 - ⁿ Frequently, global constant errno will be set
 - " Use *perror("error description");* to see what the error was
- Don't worry about architectures other than x86.
- Don't worry about compiling the shell in vmware
 - n Compile on spinlock, transfer executable to vmware
- n Q: "warning: implicit declaration of xyz" -- ???
 - A: Check include files

11