CSE 451 Section 2:

Processes, the shell, and system calls

Homework 1 highlights

- (1.7.) Availability is typically ensured through replication
- (1.11.) Memory bus contention can occur
- (2.14.) User: Your arguments +
  VMs protect from compromised software
- (2.15.) Asynchronous communication: harder to program with, but does not interrupt.

Project 1

- Teaches you:
  - how to handle processes
  - how to build & run Linux in VMware
- Two main parts:
  - Write a simple shell in C
  - Add a simple system call to Linux kernel
- Due: Wed., Oct 15, 11:59pm
  - Electronic turnin: code + writeup

Project groups

- Please form groups of 3
  - Send me emails with groups by Tues., Oct. 7
  - After that, I will assign you to random groups
  - You can change groups for next projects
- Only one person / group submits
- You can use CVS
  - Instructions of how to set it up on forkbomb and use it are on website

The shell

- What is it?

The shell

- "A program that works with the operating system as a command processor, used to enter commands and initiate their execution."

  -- *American Heritage ® Dictionary of the English Language*
- Examples of shells:
  - UNIX: bash, csh, …
  - Windows: command prompt

The UNIX shell

- Internal (built-in) commands
  - Execute routines embedded in the shell
  - Manage state of the shell (e.g., current working directory, environment variables)
  - Examples?
- External commands
  - Examples?
- How can you tell external from internal?

7

Other UNIX shell capabilities

- Redirect stdin / stdout / stderr

```
# ./my_parser < logfile > outfile 2> errfile
```

- Background execution of process

```
# ./my_parser < logfile > outfile 2> errfile &
```

- Command pipelines

```
# ps -ef | grep java | awk '{print $2}'
```

8

The CSE451 shell

- Print out prompt
- Accept input
- Parse input
- If built-in command
  - do it directly
- Else spawn new process
  - Launch specified program
  - Wait for it to finish
- Repeat

```
CSE451Shell% /bin/date
Fri Jan 16 00:05:39 PST 2004
CSE451Shell% pwd
/root
CSE451Shell% cd /
CSE451Shell% pwd
/
CSE451Shell% exit
```

9

CSE451 Shell Hints

- In your shell:
  - Use *fork* to create a child process
  - Use *execvp* to execute a specified program
  - Use *wait* to wait until child process terminates
- Useful library functions (see man pages):
  - Strings: *strcmp*, *strncpy*, *strtok*, *atoi*
  - I/O: *fgets*
  - Error report: *perror*
  - Environment variables: *getenv*

10

System Calls

- What's a system call?
- Examples?
- How do system calls compare to library calls?

11

System calls & library calls

- System call
  - Using some OS service
  - Process/Signal/File/Network/IPC/…
- Library call
  - Not using any OS service
  - Provide a high level interface for OS service
- What happens when we call
  - strncpy(3) ?
  - fgets(3) ?

12

Project 1: Adding a System Call

- Add *execcounts* system call to Linux:
  - Purpose: collect statistics
  - Count number of times you call *fork*, *vfork*, *clone*, and *exec* system calls.
- Steps:
  - Modify kernel to keep track of this information
  - Add *execcounts* to return the counts to the user
  - Use *execcounts* in your shell to get this data from kernel and print it out.

13

Example of execcounts

```
CSE451Shell% execcounts clear
CSE451Shell% cd /
CSE451Shell% pwd
/
CSE451Shell% date
Wed Sep 29 16:52:41 PDT 2004
CSE451Shell% time
Usage: time [-apvV] [-f format] [-o file] [--append] [--verbose] […]
CSE451Shell% execcounts
Statistics:
Fork:                    3      27%
Clone:                   0      0%
VFork:                   0      0%
Exec:                    8      72%
CSE451Shell% exit
```

14

Programming in kernel mode

- Your shell will operate in user mode
- Your system call code will be in the Linux kernel, which operates in kernel mode
  - Be careful - different programming rules, conventions, etc.

15

Programming in kernel mode

- Can't use application libraries (e.g. libc)
  - E.g. can't use printf
- Use only functions defined by the kernel
  - E.g. use printk instead
- Include files are different in the kernel
- Don't forget you're in kernel space
  - *You cannot trust user space*
  - E.g. unsafe to access a pointer from user space directly

16

Kernel development hints

- Best way to learn: read existing code
- Use grep –r *search_string **
- Use LXR (Linux Cross Reference): *http://lxr.linux.no/*

17

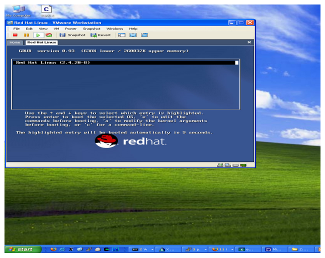
Computing Resources

- Develop your code on forkbomb
- Test your code on VMware PCs in 006
- Do not use attu

18

VMWare

- Software simulation of x86 architecture
- Run an OS in a sandbox
  - Easily reset to known good state


Red Hat Linux - VMware Workstation
File Edit View VM Power Snapshot Windows Help
Snapshot Revert
Home
Red Hat Linux
redhat
start

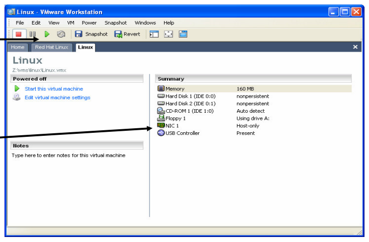
```
GRUB  version 0.93  (638K lower / 260032K upper memory)

 Red Hat Linux (2.4.20-8)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, 'a' to modify the kernel arguments
    before booting, or 'c' for a command-line.

 The highlighted entry will be booted automatically in 9 seconds.
```

19

Using VMWare

Power on/off, reset

VMWare config
Don't change!


Linux - VMware Workstation
File Edit View VM Power Snapshot Windows Help
Snapshot Revert
Home
Red Hat Linux
Linux
Start this virtual machine
Edit virtual machine settings

Linux
Z:\vms\linux\Linux.vmx

Powered off

Summary

Memory	160 MB
Hard Disk 1 (IDE 0:0)	nonpersistent
Hard Disk 2 (IDE 0:1)	nonpersistent
CD-ROM 1 (IDE 1:0)	Auto detect
Floppy 1	Using drive A:
NIC 1	Host-only
USB Controller	Present

Notes
Type here to enter notes for this virtual machine

- All disks are nonpersistent
  - *Powering off loses your changes!* Use "shutdown –r now" instead

20

Linux && VMware

- There is only one user: *root* (password: *rootpassword*)
- You will need to:
  - Build a kernel image on forkbomb
  - Transfer it to Linux running inside VMware (you can use scp from the hosting OS)
  - Boot your new Linux kernel in VMware
- Instructions at:
  http://www.cs.washington.edu/education/courses/451/08au/projinfo.htm

21