

Signals

Process-level interrupt requests

Dozens of them, use "kill -1" to list them

They can be sent

among processes as a means to communicate

by terminal to kill, interrupt, suspend processes

by kernel when encountering e.g., division by zero

by kernel to notify. e.g., data arrived on an I/O channel

Upon Receiving a Signal

• A process can "catch it", i.e., designate a signal handler routine to handle it

- Handler is called. Upon completion, resume (continue) process execution

• A process can also request to block (and then unblock) or ignore signals.

• Otherwise, kernel takes default actions on behalf of the process

- Generate core dump, or terminate the process

Core dump: a process' memory image, for debugging

Stony Brook University CSE/ISE 311: Systems Administration **Common Signals** Name Description 2 INT Interrupt (when type ctrl-C) 3 QUIT Quit 9 **KILL** Kill 11 SEGV Segmentation fault 15 **TERM Software termination**

The Kill Command: Send Signals

• Can send any signals to a process by process owner or the superuser

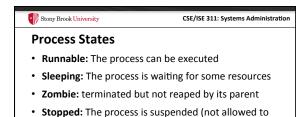
\$kill 8021 8021 is the PID

• Default is the SIGTERM, i.e., kill -TERM

• SIGTERM may not always terminate a process, kill -9 8081 sends SIGKILL

- SIGTERM may be blocked by a process

- SIGKILL is a signal that can't be blocked by processes



execute) or traced

Use the "ps" command to view a process' state

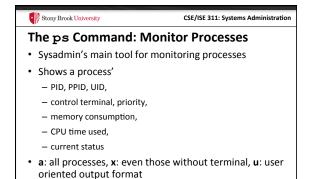


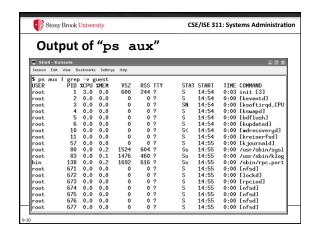
Nice and Renice: Scheduling Priority

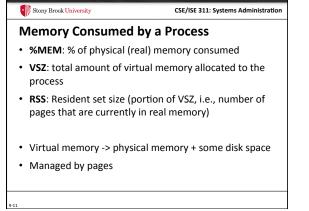
- Kernel does process scheduling: which one do I run next among the Runnable processes?
- · Process "niceness" affects the scheduling priority
 - A high nice value means a low priority
 - A low nice value means a high priority
 - In Linux, the range is [-20, 19]
- Owner of a process can increase its nice value but cannot lower it

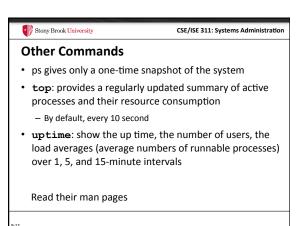
\$nice +19 ./myjob10 starts myjob10, and sets it to the lowest priority

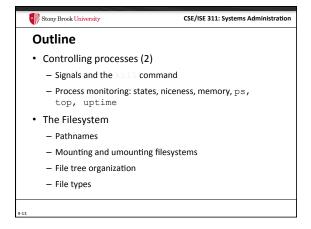
9-8

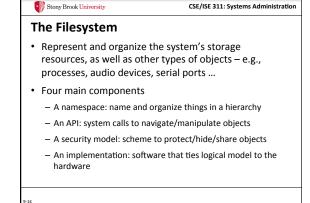












Pathnames

• The filesystem is a single unified hierarchy that starts at the directory /, and continues downward through subdirectories

– /: the root directory

• Pathname: the list of directories that must be traversed to locate a file plus that file's filename

– Absolute paths: start from root. E.g., /tmp/foo

– Relative paths: start from current directory. E.g., cse311/A1

– Terms pathname, filename, path are interchangeable

Pathnames (cont'd)

• Filesystem can be arbitrarily deep

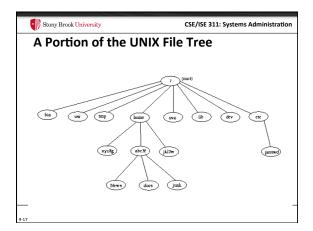
• Each pathname must be <= 255 characters

- For longer ones, cd to an intermediate directory first, then use a relative pathname

• Filenames

- Must not contain slash "/" character

- Spaces are permitted, though not recommended. E.g.,
\$less "My excellent file.txt"



• Smaller filesystems – each consists of one directory and its subdirectories and files

• Smaller filesystems are attached to the tree with the "mount" command

- Mount maps a directory in the tree (called mounting point) to the root of the newly attached filesystem

- \$mount /dev/sda4 /users install the filesystem stored on the disk partition /dev/sda4 under the path / users.

- To see the filesystem content, use ls /users

CSE/ISE 311: Systems Administration

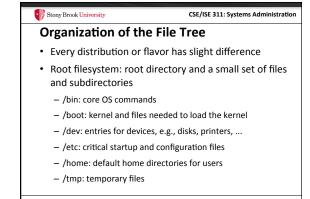
Stony Brook University



• The filesystem can not be busy, i.e., no open files or processes with current directories located there

9-19

Stony Brook University



CSE/ISE 311: Systems Administration

More Standard Directories

- /lib: libraries, and parts of the C compiler

- /mnt: temporary mount points for removable media

- /proc: information about all running processes

- /root: home directory of the superuser

- /usr/bin: most commands and executables

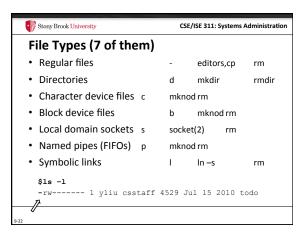
- /usr/include: header files for C compiler

- /usr/lib: more libraries

- /usr/sbin: less essential commands for sysadmins

- /var: log files, accounting info; change rapidly

- ...



File Types (cont'd)
Regular files: a series of bytes. Text, data, executable, libraries, etc.
Directories: "." refers to itself, ".." refers to its parent directory. \$cd... go to parent dir
Device files: used for hardware, peripherals.

Characterized by two numbers: major and minor device numbers. Major device number identifies a device driver. Minor tells the driver the actual unit.
E.g., the first serial port /dev/tty0 has 4,0

CSE/ISE 311: Systems Administration

