

# CSC252: Exam 1 - Review

Exam is on **D2L**, in the classroom during class time  
(you *MUST* be in the classroom and take the exam on a classroom computer)  
exam format: true/false, multiple choice, written response

## Resources

- Powerpoints:
  - 1-Intro, 2-BasicCmds, 3-Files, 4-Shells, 5-BashShell
- Examples: /export/home/public/carelli/csc252/Examples
  - Intro, Files, Shells, Bash Shell
- book, HW #1-4, lectures

## General Information

- Operating System
  - allows high level, consistent, user/machine interaction
  - manages low level interaction with the hardware - machine independent
- Unix versions: System 5 (AT&T), BSD (Berkeley), Linux
- Linux
  - open source Unix variant for x86 machines
  - Shell, applications, compilers, ... from Gnu
  - Kernel from Linus Torvalds
  - Many variants
- Unix concepts
  - everything represented as a file
  - **processes** do the work
  - Divided into two programs
    - **kernel** – machine specific, low level interactions. interrupt driven (resides in memory)
    - **shell** – user interface, common across Unix versions
  - programs and shell interact with kernel through **system calls**
- Kernel
  - data structures
    - *process table* and *open file table*
  - modes
    - System mode – unlimited system access
    - User mode – limited access

## Commands

- Run in a **terminal session** (shell)
  - syntax: `command [options] argument(s)`
  - options typically a dash followed by a single character ... `ls -l`
  - can be combined ... `ls -lt`
- command types
  - built-in
  - alias
  - executable file (system or user supplied)
    - search through directories listed in the PATH variable
- info on command
  - `man` – for general reference
  - `help` – for built-in's
- accounts
  - have id's (numerical) – see with **id** command
  - root (privileged, superuser)
  - other system accounts
    - `daemon`, `bin`, `nobody`, `lp`, ...
  - User accounts (unprivileged)
- many commands available in Unix (will not list)

## Files

- organized into **directories** hierarchically
  - starting at /
    - system binaries at `/bin` (and `/usr/local/bin`)
    - user accounts under `/home`
- directory **paths**
  - absolute (full path starting at /)
  - relative (path beginning at present location)
- Special directories: `~` `.` `..`
- `ls -l` shows extensive file info
  - permissions for u (user/owner), g (group), o (other)
  - file type (d, l, s, ...)
  - # of links, size, group, mod date, name, ...
- `chmod`, (`chgrp`, `chown`)

- links - hard vs soft
  - hard – same inode and properties
  - soft
    - “l” before permissions
    - “->” showing associated file
- file related commands: HW2
- Disk storage
  - sectors, clusters, tracks
  - partitioning and formatting
  - boot block
- inodes
  - kernel data structure for a file
  - contains all file info except name and data
  - contains list of sectors where file is stored
  - identified by an inode number (ls -i)
- File management
  - file descriptor table (process level)
  - file table (all open files, system level, in the kernel)
  - file table indexes into the inode table
- Devices
  - appear as files to Unix
  - block devices – send/receive “blocks” of data
  - character devices - send/receive individual characters
  - major (device type) and minor (specific inst) device numbers
- Raid disks
  - enable larger-effective disks and/or redundancy
  - stripping and parity
- Buffering
  - buffer pool (in memory) - *freelist* of available buffers
  - LRU (least recently used) is removed
  - sync – flush the buffers to disk

## Shells

- user interface to OS
- many versions: bash, sh, tcsh, dash, ksh, zsh, ...
- Environment variables
  - PATH, HOME, USER, SHELL, ...

- commands: env, set, ...
- metacharacters
  - file redirection: > >> < | (pipe)
  - file substitution: \* ? [...] ^ \$
  - command separator: ;
  - command substitution: ` (backtic)
- standard input, standard output, standard error (output on “2”)
  - null device: /dev/null (“bit bucket”)
- shell scripts
  - #!(path to shell)

## Bash

- login vs. nonlogin shells
- startup files
  - login: /etc/profile .bash\_profile .bash\_login .profile
    - enter password
  - nonlogin: .bashrc
    - a subshell
- bash shell scripts
  - #!/bin/bash
  - must be executable!
  - # comments
- execution of script in *current shell*
  - with dot (.)
  - with *source* command
- shell variables
  - assign with = (also *declare* command)
  - \$ to get stored value (\$VARIABLE)
  - arrays use brackets: []
  - unset
- single vs. double quotes
  - values of embedded shell variables get substituted inside double quoted strings
  - single quote disables variable substitution
- backslash (\) metacharacter escape

- command substitution
  - spawns a *nonlogin* subshell
  - subshell execution happens first
  - Two mechanisms
    - ` (backtic) quoted command
    - \$()
- aliases
  - alternate name for a command
  - embedded command substitution:
    - in single quotes occurs when executed
    - in double quotes occurs when created
  - unalias to remove
- expressions
  - arithmetic (( expr ))
  - drop \$ from variable names
- testing and Boolean operations
  - test constructs:
    - test command
    - [ expression ]
    - [[ expression ]] - (part of bash - not a command!)
  - arithmetic: -lt, -gt, -eq, ....
  - string: <, >, ==, ...
  - file: -f, -d, -e, ..
  - if-then- fi
    - also elif
- control statements
  - while-do-done
  - for-do-done
  - until, case, select, ...
  - break and continue
- command line arguments
  - positional parameters
  - \$1, \$2, \$3, ...
  - \$# (number)
  - \$\* or @\$ (all)
  - \$0 (command)
- functions
  - definition, arguments, return
  - no scoping (except for cmd line args)