

CST8177 – Linux II

bash startup files
Linux/Unix files
stty

Topics

- ▶ midterms
- ▶ bash startup files
- ▶ stty

Configuring Bash Behavior

- ▶ We customize our shell behavior by

- setting environment variables, for example,

```
export PATH=/bin:/usr/bin:/sbin
```

- setting aliases, for example

```
alias ll="ls -l"
```

- setting shell options, for example,

```
shopt -s failglob or shopt -s dotglob
```

- setting shell options, for example,

```
set -o noclobber
```

we make these customizations permanent using bash startup files

Bash Startup Files

- ▶ http://teaching.idallen.com/cst8207/13f/notes/210_startup_files.html
- ▶ `~/.bash_profile` is sourced by your login shell when you log in
 - the things we set up here are done only once when we log in
 - `export`-ed variables here are inherited by subshells
 - we source `~/.bashrc` here because login shells do not source it
- ▶ `~/.bashrc` is sourced by each non-login subshell, interactive or not
 - here we set up things that are not inherited from the login shell
 - inside this file, at the top, we check whether it's an interactive or non-interactive shell:

```
[ -z "${PS1-}" ] && return
```
 - we set aliases in this file
 - we set options configured with `shopt` and `set` in this file

Startup File Sequence

- ▶ When a login shell starts
 1. execute commands from `/etc/profile`, if that file exists
 2. execute commands from the first of these that is readable (in order):
 1. `~/.bash_profile`
 2. `~/.bash_login`
 3. `~/.profile`
- ▶ When a login shell exits
 1. read and execute commands from the file `~/.bash_logout`, if it exists

Startup File Sequence (cont'd)

- ▶ When an interactive non-login shell starts
 1. execute commands from `~/.bashrc`, if that file exists
- ▶ The `--rcfile file` option specifies that file should be used instead of `~/.bashrc`

System Wide Shell Configuration

- ▶ The system administrator can configure the default shell environment for all users
- ▶ Configuration in `/etc/profile` applies to all users on the system
- ▶ The files in `/etc/skel/` are copied to newly created user accounts (can give new users a default copy of `.bash_profile` and `.bashrc`)

Non-Interactive Shells

- ▶ The bash process used to execute a shell script is non-interactive
- ▶ `stdin` and `stdout` not connected to a terminal (more details in bash manpage)

.bashrc versus .bash_profile

- ▶ .bash_profile is loaded once by a login shell
- ▶ .bashrc is loaded by non-login shells
- ▶ There are cases where there never is a login shell, for example

```
ssh remote-server.com <some_command>
```

- ▶ So the method we'll use in this course:
 - .bash_profile does nothing except load .bashrc
 - .bashrc keeps track of things that should be done only once

.bashrc

```
[ -z "${PS1-}" ] && return
if [ "${_FIRST_SHELL-}" = "" ] ; then
    export _FIRST_SHELL=$$
    export PATH="$PATH:$HOME/bin"
    export LC_ALL=en_CA.UTF-8
    export LANG=en_CA.UTF-8
    # here we put things that
    # should be done once
fi
# here we put things that need to be
# done for every interactive shell
```

.bash_profile

Contains just one line:

```
source ./ .bashrc
```

New Commands: stty

- ▶ recall the effect of these control characters:
 - ^Z suspend the current foreground process
 - ^C terminate the current foreground process
 - ^D end of file character
 - ^U kill character to erase the command line
- ▶ these are actually properties of the terminal
- ▶ they can be set with the stty command
- ▶ `stty -a` : print out the current tty settings
- ▶ `stty susp ^X` :(that's a caret ^, shift-6 on my keyboard, followed by capital X) means set the susp character to CTRL-X instead of CTRL-Z

stty (cont'd)

- ▶ if you accidentally dump the contents of a binary file to your screen, and all the control characters reconfigure your terminal on you, you can reset it to sane values with

```
stty sane
```