COMPSCI 1XC3

L03 - Week 3:2

Processes and the Enviornment

Today’s Agenda

- Processes
- Enviornment
- Vi
- Check-off and Attendance

> **Labs = Practice & Learning, not testing. Mistakes are expected and welcome!**

1. Process

What are Processes?

- Running Programs on your system which can launch other programs
- Each process has a unique ID called **PID**
- Parent processes can launch child processes that belong to them
- OS tracks CPU usage, memory and resource allocation

Some commands related to processes

View

- `ps` → View all current processes in your shell
  - `-a` → show processes for all users
  - `-u` → display process's user/owner
  - `-x` → show processes not attached to a terminal

Monitoring

- `top` → Dynamically displays processes in order of CPU activity
- Updates every 3 second by default
- Type `q` to exit

Viewing Current Jobs

- `jobs` → List all active jobs with their status (running or stopped)

Foreground Jobs (Default)

- Default behavior when you run any command
- Shell blocks and waits until the job completes
- You cannot type new commands until it finishes
- For example `xlogo` starts it in foreground
- **How to control:**
  - `fg %1` → Bring job 1 to foreground
  - `Ctrl+Z` → Pause current foreground job (returns to shell)

Background Jobs (Run independently)

- Shell returns immediately - you can run more commands
- Multiple background jobs run simultaneously without blocking each other
- `xlogo &` starts it in background
- **How to control:**
  - `bg %1` → Resume paused job 1 in background

Terminating Processes

- `kill -signal PID` → send a kill signal to a process with `pid`
- `killall` → sends signals to multiple processes
- `shutdown` → Shut down or reboot the system

Install Xlogo (Will need it for Check-Off)

- Just do `$ sudo apt install x11-apps # One-time setup`
- Running `xlogo` launches the process

2. The Enviorment

What are Enviornment Variables?

Think of environment variables as **labeled containers** that store information your shell needs:

- **Environment variables** → Global, available to all programs you run
- **Shell variables** → Local, only in your current shell session
- Access them with `$VARIABLE_NAME` syntax

Real examples:

- `$USER` → your username
- `$HOME` → your home directory (/home/username)
- `$PATH` → where shell looks for commands

Examining Enviornment Variables

- `printenv` → Display all environment variables
- `set` → Display all variables (environment + shell)
- `echo $VARIABLE` → Display specific variable value
- `alias` → Display aliases

Creating and Exporting Variables

Step 1: Create (local to shell)

```
myVar="Hello World"
echo $myVar          # → Hello World
```

Problem: Other programs can't see it

Step 2: Export (make global)

```
export myVar
printenv myVar       # → Hello World (now visible to ALL programs)
```

Export makes variables available to child processes

Modify the PATH Variable

Problem: You create scripts in `~/bin/` but shell says "command not found"

```
# Check current PATH
echo $PATH
# For eg /usr/bin:/bin:/usr/local/bin

# Append your bin directory
PATH=$PATH:$HOME/bin

# Export to make permanent for child processes
export PATH

# Verify
echo $PATH

# Now your scripts work from anywhere
```

Now executables in `~/bin` are accessible from anywhere

Make Changes Permanent (.bashrc)

Temporary changes disappear when shell closes. Add to `~/.bashrc`

Step 1: Open `.zshrc` (Mac) or `.bashrc` (Linux/WSL):

```
nano ~/.bashrc
```

Step 2: Add the PATH:

```
export PATH="$PATH:$HOME/bin"
```

Step 3: Reload

```
source ~/.zshrc
```

3. Vi

Why learn vi?

- Core Unix/Linux text editor
- Available on almost all systems
- Lightweight and fast
- vim is "vi improved" (usually aliased to vi)

Understanding vi modes

- **Command Mode** : Navigation and editing commands (default)
- **Insert Mode** : Type and edit text Press `i` to enter Insert mode and `ESC` to return to Command mode

Essential vi Commands (Command Mode)

- `:w` → Save file
- `:wq` → Save and quit
- `:q!` → Quit without saving
- `:set nu` → Show line numbers

Searching in vi

- `/keyword` → Search forward for keyword
- `n` → Jump to next occurrence
- `N` → Jump to previous occurrence

Typical vi workflow

1. Start: `vi filename.txt`
2. Press `i` for insert mode
3. Type your text
4. Press `ESC` to return to command mode
5. Type `:wq` to save and exit

Check Off and Attendance

Lab Week 3-2 Check-off

Task 1 (Processes)

Goal: Master process control by juggling multiple jobs and cleaning up.

Task (Figure the commands out and show them to me):

1. Start first xlogo in foreground, then pause it
2. Send paused xlogo to background
3. Start second xlogo in background
4. List jobs with PIDs
5. Kill second xlogo by PID

Task 2 (Enviornment Variables)

Goal: Create a custom environment variable and make PATH changes permanent

Task (Figure the commands out and show them to me):

1. List all environment variables, then quit
2. Create & export a variable called myCourse="1XC3 Rocks"
3. Verify the new variable works
4. Append custom PATH to .bashrc
5. Reload .bashrc and verify PATH updated

Task 3 (vi Editor)

Goal: Create, edit, save, and search in vi using proper workflow.

Task (Figure the commands out and show them to me):

1. Create new file `hello.txt` and enter insert mode
2. Type "Hello 1XC3" then save and quit
3. Reopen `hello.txt`
4. Add "Easy marks" on line 2
5. Save as `helloAgain.txt` (preserve original)