

Lecture 3

Computer Hardware and The Unix Operating System

Instructor: Ashley Gannon

ISC3313 Fall 2021



Components



Memory Cell

- The fundamental unit of computer memory is a memory cell.
- This is an electronic circuit that stores one binary digit (*bit* for short) of information.
- This bit represents a logical state with one of two factors - 0 or 1, on or off, high or low voltage.
- The value of a bit is stored until it is changed by the set/reset process, and can be accessed by reading the memory cell.

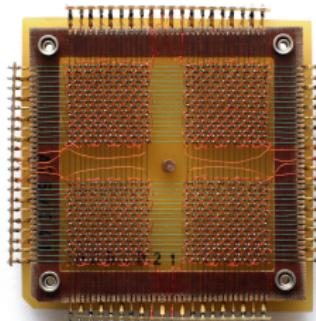


Figure: This 32x32 core memory plane stores 1024 bits of information



RAM

- The RAM consists of multiple memory cells.
- Each memory cell is uniquely identified by its memory address.
- The first memory address always starts at 0, the last memory address depends on the amount of memory installed.



Figure: Example of writable volatile random-access memory



Central Processing Unit

- The RAM memory is under control of the CPU
- The CPU can store a value in a specific memory cell in the RAM
- The CPU can recall the stored value when prompted.



The Unix operating system



What is Unix?

The Unix operating system is a set of programs that allow the user to communicate with the computer.



What is Unix?

The Unix operating system is a set of programs that allow the user to communicate with the computer.

The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or the kernel.



What is Unix?

The Unix operating system is a set of programs that allow the user to communicate with the computer.

The computer programs that allocate the system the system resources and coordinate all the details of the computer's internals is called the operating system or the kernel.

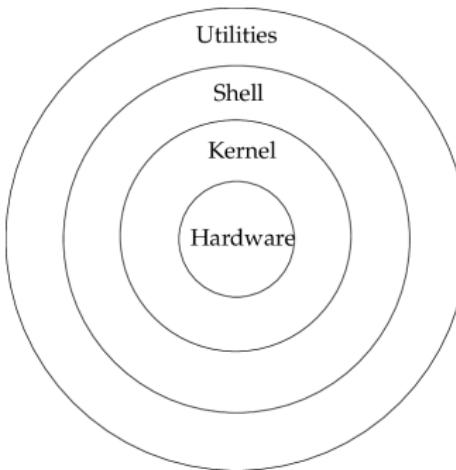
Users communicate with the kernel through a program known as the shell. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.



Unix framework

The main concept that unites all the versions of Unix is the following four basics:

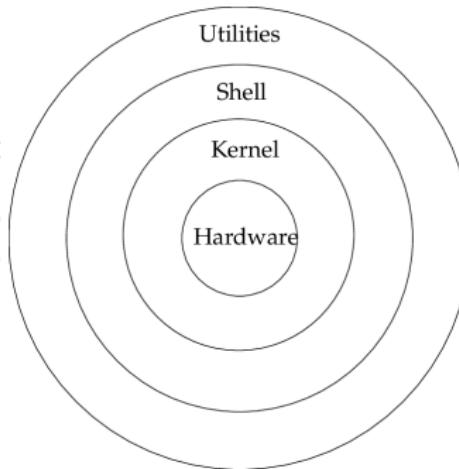
- 1. Kernel:** The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.



Unix framework

The main concept that unites all the versions of Unix is the following four basics:

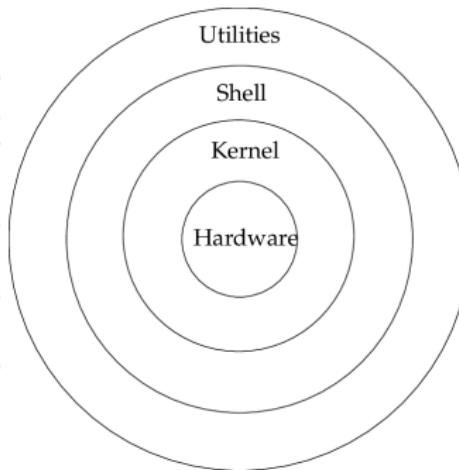
2. **Shell:** The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands.



Unix framework

The main concept that unites all the versions of Unix is the following four basics:

3. Commands and Utilities: There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.



Unix framework

The main concept that unites all the versions of Unix is the following four basics:

4. Files and Directories All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the filesystem.

```
ag12s@DESKTOP-1FNKE80:~$ ls
bin  boot  dev  etc  home  init  lib  lib64  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
```



Setting up the Bash Shell on Windows



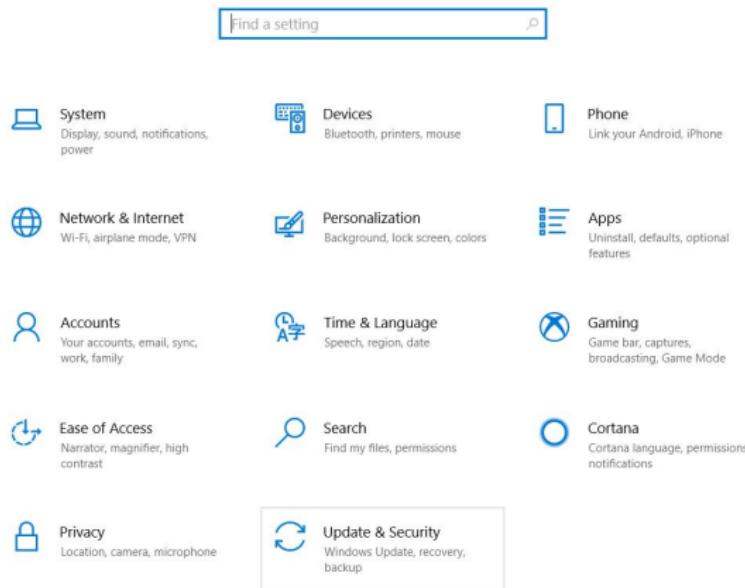
Setting up the Bash Shell on Windows

1. Navigate to settings.



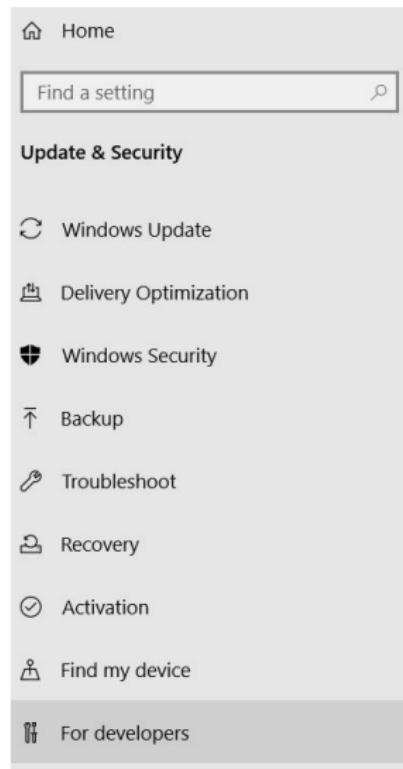
Setting up the Bash Shell on Windows

2. Click "Update and Security"



Setting up the Bash Shell on Windows

3. Select "For Developers" in the left column.



Setting up the Bash Shell on Windows

4. Select "Developer Mode"

The screenshot shows the Windows Settings application. On the left, there's a sidebar with various options: Home, Update & Security, Windows Update, Delivery Optimization, Windows Security, Backup, Troubleshoot, Recovery, Activation, Find my device, and For developers. The 'For developers' option is highlighted. The main pane is titled 'For developers' and contains the following text: 'These settings are intended for development use only.' with a 'Learn more' link, 'Developer Mode' (with a description 'Install apps from any source, including loose files.' and a 'On' toggle switch), 'Device Portal' (with a description 'Turn on remote diagnostics over local area network connections.' and a 'Off' toggle switch), and 'Device discovery' (with a description 'Make your device visible to USB connections and your local network.' and a 'Off' toggle switch). A note at the bottom states: 'Note: This requires version 1803 of the Windows 10 SDK or later.'

Home

Find a setting

Update & Security

Windows Update

Delivery Optimization

Windows Security

Backup

Troubleshoot

Recovery

Activation

Find my device

For developers

For developers

These settings are intended for development use only.

Learn more

Developer Mode

Install apps from any source, including loose files.

On

Device Portal

Turn on remote diagnostics over local area network connections.

Off

Device discovery

Make your device visible to USB connections and your local network.

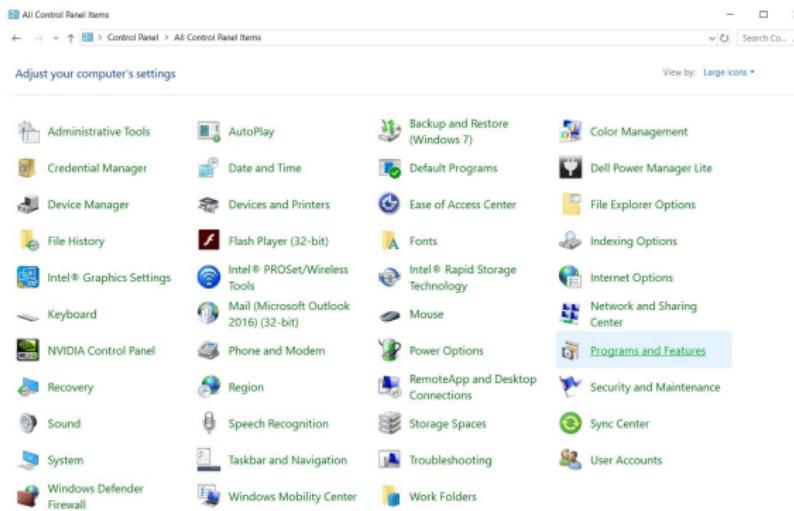
Off

Note: This requires version 1803 of the Windows 10 SDK or later.



Setting up the Bash Shell on Windows

5. Go to your control panel, select "Programs and Features"

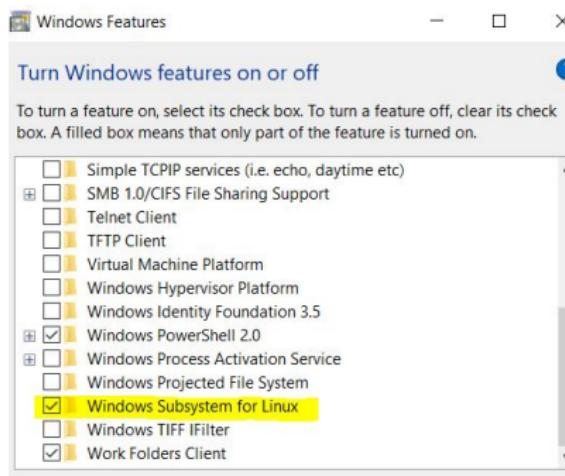


Setting up the Bash Shell on Windows

6. Click "Turn Window's Features On or Off"



7. Check the box next to "Windows Subsystem for Linux"

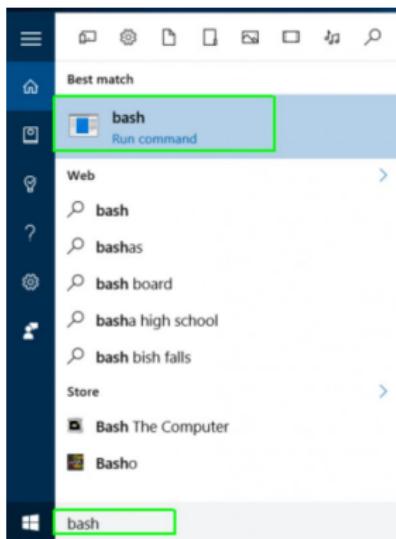


8. Restart your computer.



Setting up the Bash Shell on Windows

- Now that your computer has booted back up, search "Bash" in the search box and select the Run command.



- Type "y" and hit Enter when prompted to install Ubuntu. The system will then take a few minutes to install Ubuntu in the command prompt window.

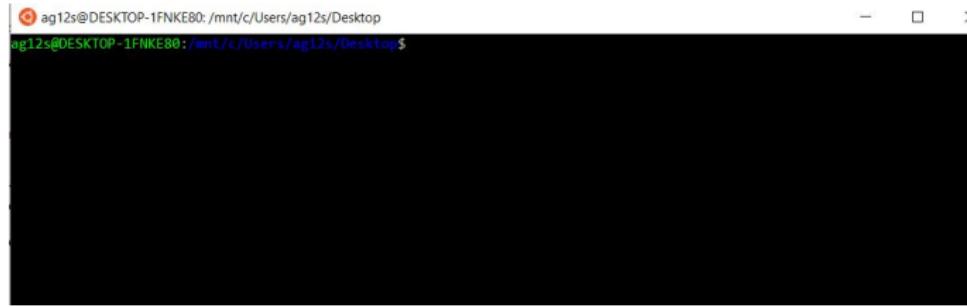
- Create a username and password when prompted. NOTE: You will NOT see anything being typed for your password, but it is typing.



Starting Your Shell in Your User File

- 1 Open the Bash shell
- 2 Type "cd ~". This will bring you to the home directory.
- 3 Type "vim .bashrc". This will open up a scary looking code in vim.
- 4 Hit the down arrow until you make it to the bottom of this code.
- 5 Type i - this is the short key for insert.
- 6 Make a new line, and on the new line type "cd /mnt/c/Users/YOURUSERNAME"
- 7 Hit the esc key
- 8 Type ":wq"
- 9 Restart Bash

I have mine set up to start on my Desktop.

A screenshot of a terminal window titled 'Terminal' on a Mac OS X desktop. The window shows a command-line interface with the following text:

```
ag12s@DESKTOP-1FNKE80: /mnt/c/Users/ag12s/Desktop
ag12s@DESKTOP-1FNKE80:/mnt/c/Users/ag12s/Desktop$
```

The terminal window has a dark background and light-colored text. The desktop background behind the window shows a blurred image of a city skyline at night.