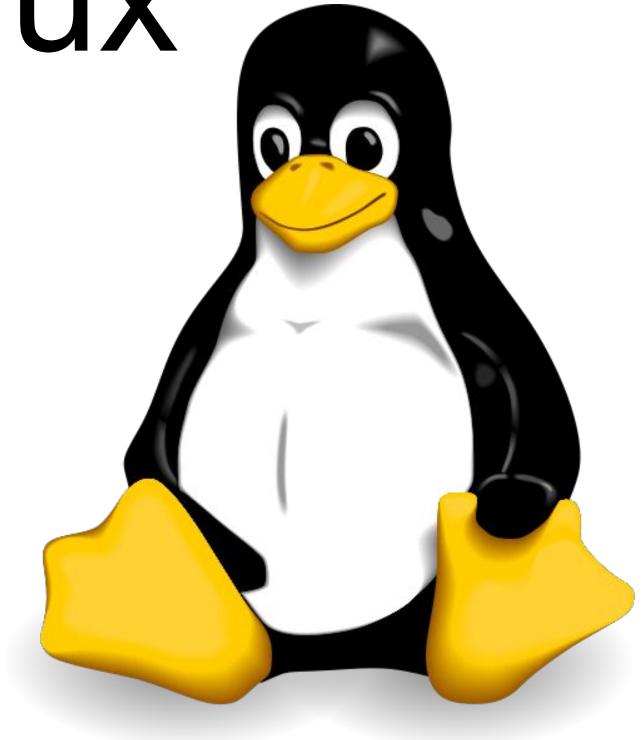
Intro to Linux



Setting up

- If you do not have Linux on your machine you can use the following online terminal

  http://www.tutorialspoint.com/execute_bash_online.php

- You can view this presentation at:

  https://goo.gl/eljsja

What is Linux?

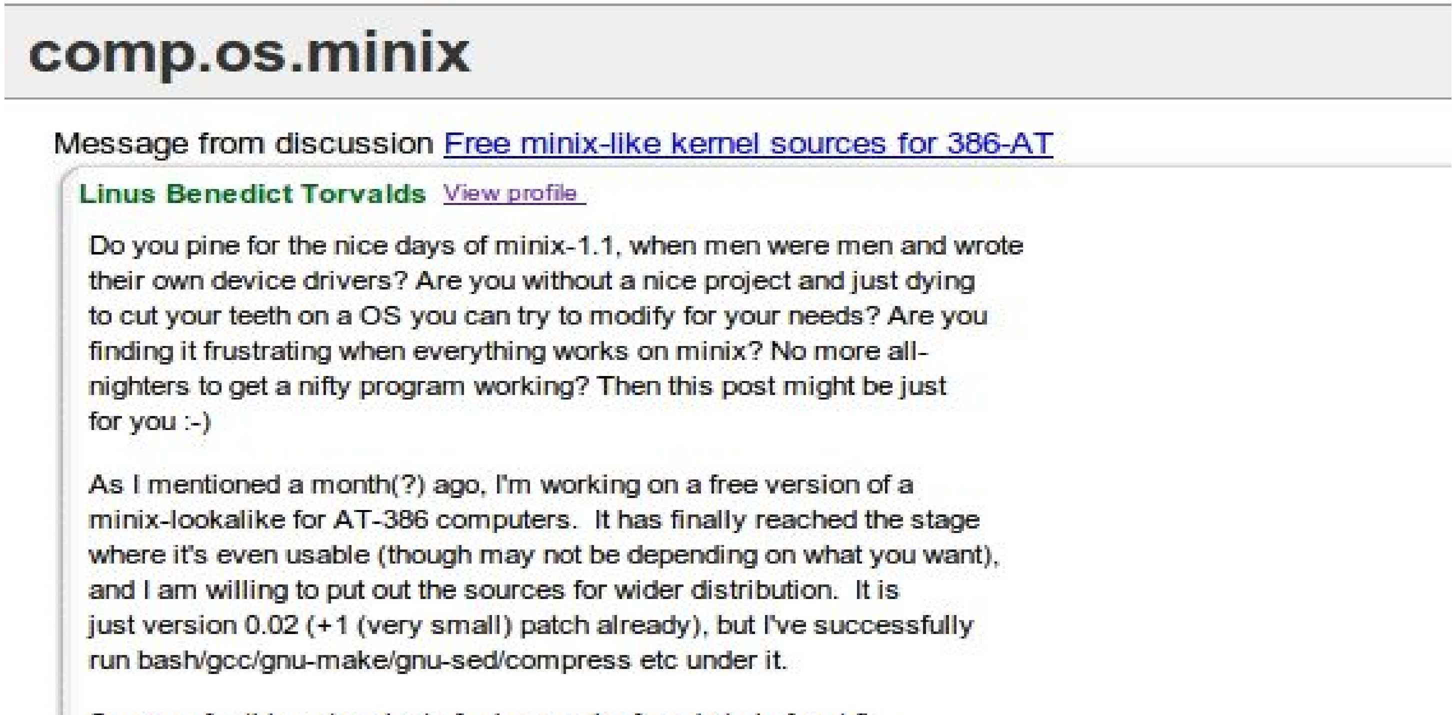
- Linux is an open source Unix-like computer operating system created by Linus Torvalds which was originally designed for the 386 processors created by Intel but has since then been ported to nearly every computer architecture, List of Architectures Supported by Linux
- Today Linux runs on a wide variety of hardware from mobile phones (Android uses Linux), routers, televisions, consoles, toasters, desktop computers, most servers, and nearly every supercomputer in the world[1]

What is Linux?

- While Linux actually refers to the **kernel** created by Linus, it commonly is used to describe the collection of tools, software, and interface that make up a usable system
- Linux is a free and open source **(FOSS)** operating system which uses the GNU General Public License **(GPL)** in order to implement a "Copyleft" licenses that requires it to remain open, free, and accessible to anyone

What is Linux?

- Today, Linux is developed by thousands of contributors and experts from around the world (such as one of my previous professors) and there are large for-profit companies such as RedHat who actively contribute to it, RedHat video
- Announcement of Linux



comp.os.minix

Message from discussion Free minix-like kernel sources for 386-AT

Linus Benedict Torvalds View profile

Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)

As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Getting Around With the CLI

- **Paths in Linux -** Linux uses a **/** for path names, don't forget this, **~** is used to reference your home directory
- **cd -** "change directory", used to change directories, simply type **cd** and the name of the path to change to (**cd ~/folder)**
- **pwd** - print your current directory, displays your current path
- **ls -** show a listing of directory contents
- **mkdir -** creates a directory with the name given (**mkdir test)**

Getting Around With the CLI

- **touch -** changes date/time stamp of a file, can also create an empty file if file does not exist **(touch myfile)**
- **cp** - copies a file from source to destination (**cp file1 file2**)
- **mv -** change the location/name of a file or directory (**like cut**)
- **rm -** removes a file **(rm oldfile)**

Getting Around with the CLI

Examples:

1. Create a directory in your home folder:
   - ***mkdir <your name>***
2. Change to that directory:
   - ***cd <your name>***
3. Display your current path, check you're in test:
   - ***pwd***
4. Create some files, make as many as you like:
   - ***touch file1***
5. Make another directory inside:
   - ***mkdir test2***
6. Display a list of the contents of directory:
   - ***ls***
7. Move some files into test2 directory:
   - ***mv file1 test2***

Getting Around with the CLI

Examples:

8. Display the contents now:
    - ***ls***
9. Go to test2 directory display files:
    - ***cd test2 && ls***
10. Remove some files:
    - ***rm file1***

Some Useful Commands

- **man** - bring up the manual for any command/documentation, provide man with the program manual you need **(man mkdir)**, to exit from the manual enter **:q (quit)**
- **--help** not a command, useful argument pass it to any command to get a brief list of available commands/arguments **(mkdir --help)**
- **ln -s -** Create a symbolic link to a file, think of it as a shortcut **(ln -s /path/to/somefile shortcut)**
- **find -** seach for a file/folder based on pattern **(find /path/ -name "somefilename"),** here is a good find tutorial

Some Useful Commands

- **grep -** search for content matching a patter **(grep 'something' filename)**
- **less -** useful for handling/modifying/viewing streams of output
- **cat -** used to concatenate files, often used to output data, which is a UUOC

Some Useful Commands

Examples:

1. Display the manual for some command:
   - ***man grep***
2. Display the help for some command:
   - ***grep --help***
3. Create a symbolic link to a directory from before:
   - ***cd ~ && ln -s ~/<your name>/test2 shortcut***
4. Find a file in a directory, try some others as well:
   - ***find /etc -name "hosts"***
5. Search for a string/pattern in a file:
   - ***grep 'localhost' /etc/hosts***
6. Output the contents of some file and pass it to less, use **:q** to quit:
   - ***cat /etc/hosts | less***

Pipes, Streams, Process Management

- Pipes or pipestreams allow inter-process communication between programs and are important to the UNIX philosophy of "many small commands working together"
  - Chain commands together using streams and pipes
  - The "pipe" is done using a **|** between commands
- Commands communicate using 3 standard streams, most often redirection is used to catch and print errors (STDERR)

STDIN (0) - input
STDOUT (1) - output
STDERR (2) - errors

Pipes, Streams, Process Management

- Pipes and streams are collectively used with redirection to control input/output and streams [redirection]
- Each command is given a process id (pid) when it is executed and handles signals, signals enable control of processes such as stopping a program, sending it to the background, or setting a program to suspend [1] [2] [3]

Pipes, Streams, Process Management

Examples:

1. Send the contents of the file hosts to grep, search for local, the pipe **" | "** takes the STDOUT from find and sends it to grep's STDIN:

```
cat /etc/hosts | grep 'local'
```

2. Redirection send output of a command to a file **">" truncates** any existing data in the file, **">>" concatenates**

```
date > output
date >> output && cat output
pwd > output  && cat output
```

***notice the output was overwritten*

Pipes, Streams, Process Management

3. Redirection and streams, redirect STDERR caused by grepping a non-existant file to STDOUT

```
    grep 'sometext' wrongfile > blank      **cat the file it's empty
    grep 'sometext' wrongfile > blank 2>&1  **redirect STDERR to file
```

4. Processes and management, list processes, control processes

```
    ps -eaf          **display processes, man ps for more info
    top              **shows process, system summary CTRL-C (signal) to
quit it
    sleep 120 && echo "hi" > file    **press CTRL-Z signal to stop it
    fg 1                              **bring process to foreground
    bg 1                              **send process to background wait
for file
```

Basics of Vim (Sorry No EMACS)

- Vim short for "vi improved" is a CLI and GUI based terminal that emphasizes having all the shortcuts and commands on the main row of the keyboard
- Vim can be a difficult editor to use at first but is one of the best multipurpose editors in existence (***bias here!)***
- Vim has three primary modes, **normal mode (press ESC), insert mode (press i), visual mode (press v)**

Basics of Vim (Sorry No EMACS)

- The following is a basic introduction to vim, but you can learn a lot more from the built in vim tutor, **vimtutor** [1]

1. Opening/Creating Files: ***vim filename***
2. Entering "insert mode: *press* ***i*** ***you can enter text in insert mode*
3. Entering "normal mode": *press* ***ESC*** ***required if not entering text*
4. Saving Files: *press* ***ESC*** *then* ***:w***
5. Save and quit: *press* ***ESC*** *then* ***:wq***
6. Searching files: *press* ***ESC*** *then* ***/wordtosearch***
7. Search and replace: press ***ESC*** *then*
   :%s/wordtosearch/replacement/gc