

# Toledo Area Linux Users Group

## Introduction to \*nix

Abdulmajed Dakkak

adakkak@eng.utoledo.edu

Toledo Area Linux Users Group

January 23, 2008

# Goals of this Presentation

The goal of this presentation is to introduce you to the command line. You should be able to copy, move, list, and navigate any \*nix system by the end of this presentation. It is up to you, however, to ask me questions at any point if you want to go beyond these objectives.

# What is \*nix?

\*nix, or UNIX Like, operating systems are operating systems that are the “same” under the hood for new users. Open source \*nix’s include: \*-BSD, Linux, and open-Solaris; while closed source ones are Solaris, Apple OSX, AIX, HP-UX, and much more.

All \*nix’s have the same basic command line tools, and they differ (depending on which you compare) by what user interface they present and how they store configuration files.

# Why use \*nix?

- Open source options

# Why use \*nix?

- Open source options
- Powerful command line

# Why use \*nix?

- Open source options
- Powerful command line
- Industry standard

# Why use \*nix?

- Open source options
- Powerful command line
- Industry standard
- Bleeding Edge

# List Content of a Directory — `ls`

```
ls
```

**Lists** files in the current directory

- `ls -l` ⇒ column view
- `ls -t` ⇒ sort by time
- `ls -r` ⇒ reverse sort
- `ls -h` ⇒ print sizes in human readable form
- `ls -a` ⇒ shows “hidden” files
- `ls -rthl`

```
cp [file1] [file2]
```

**Copies** [file1] to [file2]

- `cp -r` ⇒ copy recursively
- `cp -i` ⇒ copy interactively

```
mv [file1] [file2]
```

**Moves** or **renames** [file1] to [file2]

- `mv -r` ⇒ move recursively
- `mv -i` ⇒ move interactively

## Caution

`mv [file1] [file2]` will overwrite [file2] if it exists. This also applies to `cp`.

```
rm [file]
```

**Removes** the file [file] permanently

- `rm -fr` ⇒ forcefully removes a file or directory

## Caution

`rm` will not put anything in a trash can. A removed file is **gone** forever.

Do I remember every command and all its options . . . NO. Instead, I usually type `man [command]` to know the syntax of a command, and discover new options that can be enabled.

If you do not know what command to use, then you can use the `-k` option for `man` or the command `apropos`.

# Questions?

# Advantages of CLI

Listing all files that end with `.txt`

- **GUI** — Easy
- **CLI** — `ls *.txt`

# Advantages of CLI

Listing all files that end with `.txt` and start with `a`

- **GUI** — Easy
- **CLI** — `ls a*.txt`

# Advantages of CLI

Listing all files that end with `a.txt`

- **GUI** — **Hard**
- **CLI** — `ls *a.txt`

# Other Commands

- `mkdir [directory_name]` ⇒ **Creates a directory**
- `pwd` ⇒ **Prints the present working directory**
- `cat [text_file]` ⇒ **Concatenates a file**
- `head [text_file]` ⇒ **Prints the first 10 lines of a file**
- `tail [text_file]` ⇒ **Prints the last 10 lines of a file**
- `file [file]` ⇒ **Prints the file type of [file]**
- `vi/emacs/nano` ⇒ **Text file editors**

## Other Commands Cont...

- `cal` ⇒ Prints the calendar
- `ps` ⇒ Lists current running processes
- `bc` ⇒ Arbitrary precision calculator
- `date` ⇒ Prints the date and time
- `grep [regex] [file]` ⇒ Regular expression matching of text files
- `less [text_file]` ⇒ File pager

# Questions?

FIN