

# UNIX Operating System

true multiuser operating system - each user has their own private space on the machine's hard disk and is identified by an id number

user - user name and password

user with id 0 - system's superuser/administrator; usual name - root

UNIX acts on behalf of the initiating user

multiple users can run multiple processes simultaneously

users into groups - set of users with the same class of permissions

everything is represented by **files**:

**configuration** files - user profile, system/server configuration is plain text files

-> easy backup/restore/compare configuration, remote administration

**devices** - regular file I/O operations on file system object (under /dev)

## Filesystem

```
--bin      Binaries required before mounting /usr
--etc      System wide configuration files
--home     Users' home directories
--lib      Libraries required by the system
--tmp      Temporary files. Everyone has RW access.
--usr      Programs
|  |--bin  Programs' executables
|  |--lib  Programs' libraries
|  |--local Programs that are install locally
|  |      |-bin,lib,share
|  |--share Programs' required files (e.g. docs, icons)
|  |--sbin System administration programs
|  |--src  Source files for the kernel and programs
--var      Temporary space for running programs
```

```
permissions: $ ls -la /bin/ls
-rwxr-xr-x 1 root wheel  38624  Jul 15 06:29  /bin/ls
\         /      |      |      |      \         /      |
Permissions  Owner  Owning  Size      Date of  Filename
                group      last modification
```

paths:

absolute - start from root directory ( /var/log/messages )

relative - current directory . (inside var: ./log/messages )

# File Commands

## Listing

**ls** - list files in a directory

-l: list details

-a: list hidden files

**find** <dir> - walk through file hierarchy from <dir>

-type [**dfil**]: display **d**irectories, **f**iles or **l**inks

-name str: display entries that start with str

-{max|min}depth d

## Manipulation

**touch** <file> - create an empty file named <file>

**cp** <from> <to> - copy file from one directory to another

-R: recursive copy

-p: preserve filesystem permissions and attributes

**mv** <from\_1>...<from\_n> <to> - move files from directories

-n: don't overwrite existing files

**mkdir** - create empty directory in relative path

-p: create intermediate directories as required

## Text File Processing

**cat** file\_1 ... file\_n - concatenation and print files

**less** file - display file on screen

q - exit

/pattern - search pattern in text

**echo** <string> - write arguments

**head/tail** <file> - display first/last lines

-n: number of lines

-f: newly appended lines

## Process Management

multitasking - execute many tasks at once, dividing the processor's time

Ctrl + Z - suspend the running process

Ctrl + C - kill the running process

append & after command - start at the background

## Manipulation

**ps** - see running processes

**kill** -<signalno> <pid> - send signal to process:

TERM - inform process should terminate

KILL - directly kill process

## Advanced Text Flow

2 data streams: STDOUT (1) and STDERR (2)

2 types of text flow control:

**redirect** - send a program's output to a file (>) or read from one (<)

**ps -ef > processes**

> - overwrite; >> - append

**pipe** - forward program's STDOUT to input of another program

## Pipeline

| - pipe operator

cat file | wc -l: count lines of file

find / -type d | sort: view sorted directories

cat /var/log/access\_log | grep foo | tail -n 10: see last 10 accesses of "foo"

cat /etc/passwd | cut -f1 -d':' | sort - sorted list of users

## Documentation

self-documenting system

commands/tools - manual page (arguments, inputs, outputs, interfaces)

**man** <cmd> - invokes manual for cmd