Linux Tutorial

Introduction/Tutorial on the Linux Ecosystem

Lan Nguyen
Data-Intensive Distributed Systems Laboratory

Table of Contents

Introductions

The Shell

Bash Scripting

Secure Shell

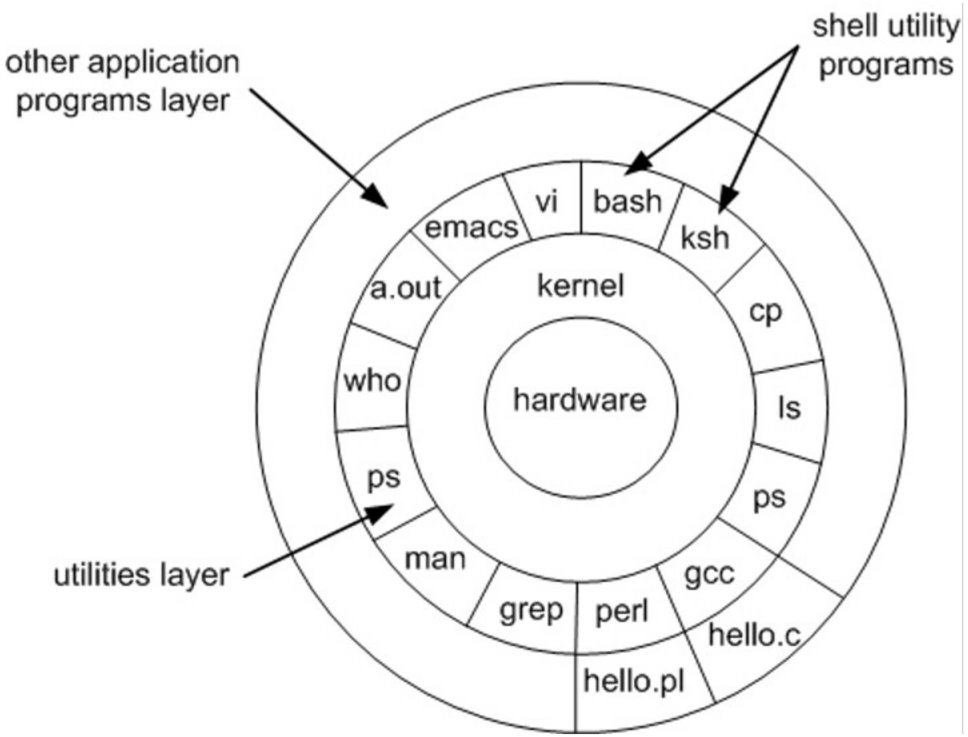
Git

Virtual Machines

Measuring Execution Time

Operating System (Overview)

- Acts as an intermediary between user and hardware
- Manages computer hardware and software resources
- Large and complex software, implemented in layers
- Variety in purpose, design and implementation
- Process, Memory/Storage, Input-Output



Source: https://homepages.uc.edu/~thomam/Intro_Unix_Text/OS_Organization.html

The Shell

- Collection of utilities, text-based interface
- Allows user to communicate with the computer
- Allows program to be started and tasks to be run
- Shell scripting
- Getting the Shell:
    - Windows:
        - Windows Subsystem (WSL) for Linux, Cygwin
    - Mac, BSD, Solaris and Linux:
        - Terminal

1	sh (Bourne Shell)	Basic shell, all UNIX systems
2	ksh (Korn Shell)	Complete high-level programming language
3	bash (Bourne Again Shell)	Combines Korn shell and C Shell (csh) **Default** in most Linux distributions
4	dash (Debian Almquist shell)	Fast, secure and minimalist shell
5	zsh (Z-Shell)	Highly interactive and improved tab completion

UNIX/Linux Commands

- File commands (ls, cd, pwd, rm, mv, cat, head)
- Process management (ps, top kill, bg, fg, jobs)
- File permissions (chmod, chown)
- System information (whoami, uname, man, du, df, free, which)
- Compression (tar, gzip)
- Network (ping, dig, wget, ssh, ifconfig, ip)
- Build and install (make, cmake, automake, dpkg, rpm, gcc)
- Shortcuts (Ctrl+C, Ctril+Z)

Introduction - https://youtu.be/SkB-eRCzWIU?t=615

Cheatsheet: - https://files.fosswire.com/2007/08/fwunixref.pdf

Image source: https://en.wikipedia.org/wiki/Linux

Bash Scripting

- “Gluing together” system calls, tools, utilities, and programs
- Automate configuration steps and file editing procedures
- Automate program iteration under variant arguments
- Best suited for system administration tasks

Additional information - https://tldp.org/LDP/abs/html/

Image source: https://www.asianscientist.com/2017/04/tech/lego-blocks-metamaterials/

Bash Scripting (Example)

```
#!/bin/bash
function clear_cache {
    sync
    sudo sh -c 'echo 3 > /proc/sys/vm/drop_caches’
}
export CLASSPATH="lib/lucene-7.1.0/build/core/classes/java/:bin"
path_hdd="/home/cerberus/data" path_ssd="/storage/data"
log="iteration.log”
echo -n "" > $log
for i in {1..10}
do
    file="dataset$(($i * 200))MB.txt”
    clear_cache
    java XSearchData $path_hdd/$file $path_hdd/terms.txt &>> $log
done
```

Image source: https://www.asianscientist.com/2017/04/tech/lego-blocks-metamaterials/

Secure Shell (SSH)

- Cryptographic network protocol
- Secure channel over unsecured network
- Client-server architecture (SSH client and SSH server)
- OpenSSH is the most popular implementation

Use cases:

- Login remote host
- Execute command on remote host
- Automatic login (Passwordless login) to remote server
- Secure file transfer
- Forwarding and tunneling ports
- Forwarding X from a remote host

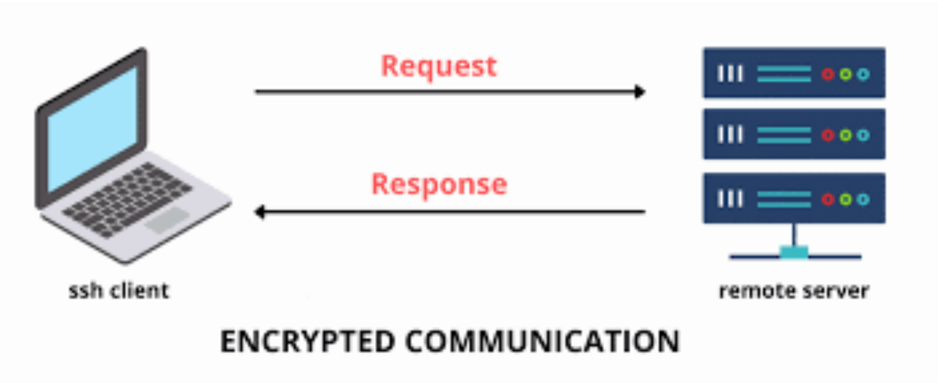


Image source: https://geekflare.com/understanding-ssh/

Secure Shell (Example)

```
localhost$ ssh-keygen
localhost$ ls –l ~/.ssh
-rw------- id_rsa
-rw-r--r-- id_rsa.pub

localhost$ ssh-copy-id <username>@<remote-host>
localhost$ ssh –i ~/.ssh/id_rsa <username>@<remote-host>

localhost$ scp ~/.ssh/id_rsa.pub <username>@<remote-host>:id_rsa.pub
<remote-host>$ touch ~/.ssh/authorized_keys
<remote-host>$ chmod 644 ~/.ssh/authorized_keys
<remote-host>$ cat id_rsa.pub > ~/.ssh/authorized_keys
```

Git

- Version-control system for tracking changes in files
- Fast, ensures data integrity
- Distributed, non-linear workflows
- Free and open-source

Use cases:

- Source code management
- Coordinate work between multiple developers
- GitHub, GitLab, Atlassian Bitbucket
- The development of the Linux kernel

Tutorial - https://docs.gitlab.com/ee/gitlab-basics/

Cheatsheet - https://www.git-tower.com/blog/git-cheat-sheet/

Image source: https://geekflare.com/understanding-ssh/

Git (Example)

```
$ git clone git@gitlab.com:cs553-spring2020/project-z.git

$ git add main.c

$ git commit –m “Updated the main function”

$ git push

$ git fetch

$ git pull

$ git merge
```

Image source: https://geekflare.com/understanding-ssh/

Virtual Machines

- Effective resource sharing, higher flexibility and an increased degree of security
- Operating systems, programming languages, compilers and computer architecture
- Process VMs: Multiprogramming, Emulators and Dynamic Binary Translators (Windows NT), Binary Optimizers, High-Level Language VMs (JVM and .Net)
- System VMs: Virtual Machine Monitors, Emulators, Co-designed Virtual Machines

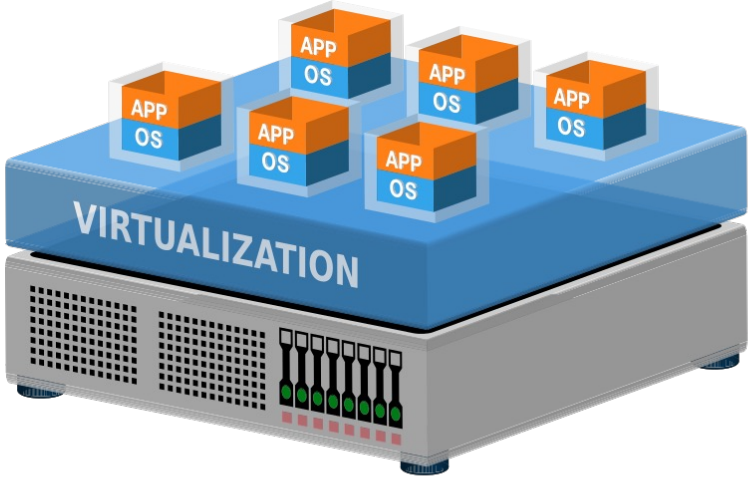


Image source: https://techgenix.com/linux-virtual-machine-vm/

System Virtual Machines

Type-1, native or bare-metal hypervisors:

- Xen
- Oracle VM Server
- VMWare ESX/ESXi

Type-2 or hosted hypervisors:

- Linux QEMU and KVM
- Oracle VirtualBox
- VMWare Workstation

Light-weight Virtualization:

- Linux LXC/LXD
- Solaris Zones
- BSD Jails
- Docker

Image source: https://geekflare.com/understanding-ssh/

Measuring Execution Time

```
$ time du –sh /home
8.4G        /home/


real        0m17.233s
user        0m0.350s
sys         0m1.850s
```

- Wall (Real) time – Total time from start to finish, including wait time (end of process quanta or waiting for I/O to complete)
- User time – Total time spent on the CPU in the user space (Other processes and the time the process spends blocked do not count)
- Sys time – Total time spent on the CPU in kernel space (Other processes and time the processes spends blocked do not count)

Image source: https://geekflare.com/understanding-ssh/

ILLINOIS
TECH