Contents

1

Linux

1.1 Resources

Books

1. Unix and Linux System Administration Handbook (Ordered)
2. The Practice of System and Network Administration

Communities

1. Superuser → https://superuser.com/
2. Server fault → https://serverfault.com/
3. Digital Ocean → https://www.digitalocean.com/community/tutorials

Sites

1. Ubuntu → https://help.ubuntu.com/
2. Tutorial Linux → https://tutorialinux.com/

Links

1. https://www.slideshare.net/kavyasri790693/linux-admin-interview-questions
2. http://simplylinuxfaq.blogspot.in/p/linux-system-admin-interview-questions.html
3. https://github.com/kylejohnson/linux-sysadmin-interview-questions/blob/master/test.md
4. https://github.com/chassing/linux-sysadmin-interview-questions#hard

1.2 Linux Facts

Mascots

1	Linux Mascot	Tux the penguin
2	BSD Mascot	Beastie the Daemon

1.3 Users, Passwords & Permissions

Users

1	Root	UID : 0, GUID : 0 (root)
2	Root Permissions	RW permissions for all files, but execute privilages can be removed
3	pseudo-users	Have a group w/ special privilages, use su <Group> to login as that group, w/ root this sets the group to the group defined
4	Adding a user	useradd <uname> (single) → newusers <batch file> (batch mode useradd). With no args a user is created with the system defaults, usually with a home dir etc.
5	Lock an Account	usermod -l <user>
6	New password	passwd <username>
7	Default file permissions	Set UMASK in /etc/login.defs (debians). Takes away the permissions
8	Change Owner & Group	chown
9	Password & login info	/etc/passwd → the hashed password itself is held in /etc/shadow
10	Change Permissions	chmod Bit mask OGA rwx
11	Delete User	userdel, removing recusively home folder and files → userdel -r

Groups

1	Wheel	Group allowing access to the sudo/su command to become another user or the superuser, for sudo this is enabled with visudo.

2	Add user to a group	usermod -a -G <group> <user> (-a only used with -G, without -a, -G makes the given groups the only additional groups he is a member of)
3	Change users primary group	usermod -g <group> <user>
4	New Group	groupadd <group>
5	All groups on system	getent group
6	chgrp	change the group ownership of a file

Sudo

1. Add a user as a sudoer by using visudo. You can specify users or groups. Groups have a % infront to seperate them from users
2. Common to have a sudo or wheel group and to give that group root permissions in visudo
3. Syntax → <user> computerAddress=(<Runas_Alias>) <Command_Alias>
4. You can use a Runas_Alias to define a semi-super user that owns a group of files or processes. Then the user can use sudo to run as that user. Same you can limit the commands that a user can run as sudo with the Command_Alias
5. to give sudo root access use: <user> ALL=(ALL) ALL → root privilages to <user> with use of sudo

1.4 Processes

Process Info

1	PID	Process ID → PID 1 is init, spawns all other ids
2	proc	In /proc → State of running processes in a virtual file system
3	Process types	user → started w/out special permissions, daemon → exist in background, kernel → execute only in 'kernel space'
4	Forked	process being started by a parent process
5	Nice	Priority level [-20 (Highest) → 19 (Lowest)] → 0 is default. Call with: nice <val> <process>, reset the priority level with: renice <new val> <PID>
6	Process Monitering	Top, ps aux, htop → good tool

Process states

RHEL Doc: https://access.redhat.com/sites/default/files/attachments/processstates_20120831.pdf

1	**R** → Runnable/Running	1. Born or forked 2. Ready to run or runnable 3. Running in user space or running in kernel space
2	**S** → Sleeping/Waiting	1. Present in main memory 2. Present in secondary memory storage (swap space on disk)
3	**D** → Blocked/Uninterruptable sleep	Very fast, unobserved, just high priority
4	**T** → Temporarily Stopped	Temporarily stopped but can be restarted
5	**Z** → Zombie	Terminated but parent process has not released it yet

Process Signals

1	kill	Send a signal to a process with: kill -s <val> (default is 15) → see man(7) signals for the signals. Defaults to -9
2	pgrep	Use user or type to find the PID of processes
3	pkill	same as pgrep but it stops the matching PID

1.5 Bash Scripting

Shell Variables

1	Set a shell variable from a program output	$(arg) or ‘<arg>‘
2	getconf	List system config variables
3	export	Allows a shell variable to be accessed by called processes

4	&&	call a command only if the proceeding one exited successfully. <command 1> && <command 2>
5	||	call a command only if the proceeding one failed. <command 1> ||<command 2>

Pipes & Redirection

1	Pipes	Sends the output of one file into the input of another → cat <filename> | grep <string>
2	Redirect	Use > to overwrite a file, >> to append. Use 1>> for STDOUT & 2>> for STDERR, use >& to redirect both.<command> < <file> send the file contents to the command

General Tools

1	curl	Tool for talking over several different protocols
2	wget	Downloads files from an address, same as curl but GNU

1.6 Maintenance

Running Jobs

1	Schedule Jobs (user)	crontab, edit using crontab -e, kept in /var/spool/cron/crontabs, also package specific cron jobs are in /etc/cron.d
2	Schedule Jobs (system)	/etc/crontab
3	at	Run a process at a specified time, accepts HH:MM
4	batch	Run a process when the load drops to a specified level
5	Job at boot	Crontab w/ @reboot

Backups

Backup Tools: http://www.admin-magazine.com/Articles/Using-rsync-for-Backups
Rsync Snapshots: http://www.mikerubel.org/computers/rsync_snapshots/

- rsync → Remote/Local, Local/Remote, & Local/Local file copying. Sends only the differences be-

tween the source & existing files in the destination

- Use: rsync <options> <source> <destination>
  - Source → Can be files → *.c, or everything in a directory <path name>/, remove the trailing slash to copy the directory.
  - To specify a remote host <computer name> use → <computer name>:<path> as the <source> or <destination>. No : means local only.
- Options:
  - -a → Archieve mode, saves symbolic links, devices, attributes, permissions, ownership, groups, and is recursive (i.e. -a == -rlptgoD).
  - -t → Transfer files, if file exists, remote-update protocol is used to update the file by sending only the differences
  - -z → Compress before sending
  - −−delete → Delete files from the recieving side if not in backup (CAUTION: run −−dry-run to see what will be removed first)
  - −−progress & -v tells you whats going on

- Backup Types → All can be done using rsync
  - Incrimental → Only record changes from last incrimental backup
  - Differential → Records changes since the last total backup
  - Replica → Just replicate the whole shebang

- Rsync for incrimental backups
  - Have a full backup <Full Backup> → rsync to a fresh loc
  - Have <Backup.0> which has all the incrimental changes
  - Make each backup look like a full backup using hard links (cp -al)

1.7 Strings & Searching

Grep

1	Description	Search for a character pattern in a string
2	Use	grep <string> <filename> → returns the lines with the character pattern <string> in file filename
3	<option> -r	Follow directories
4	<option> -n	Get the line number
5	<option> -l	Get files with the string
6	<option> -i	Ignore case

Strings

1	cut -d <delim> -f <field1>-<field2>	Break a line on a delim, then take the fields in range, c of chars, b bytes
2	sed	Stream editor, based on the original UNIX tex editor ed.
3	awk	Pattern scanning and processing language
4	python	For normal people use python w/ the re package.
5	tr	Translate, use for replacing certain strings with something else. I mean really just use python, but theoretically use this
6	tee	Put standard in to a file and to standard out, useful for logging the output while filtering

1.8 Files

Files

1	Types	7 types block special, char special, directory, normal file, symbolic link, named pipe, socket
2	diff	Get difference between 2 files or dirs
3	comm	select or reject common lines between files
4	ln -s	Create a symbolic link → sym links dont have to exist unlike hardlinks. Same as a shortcut.
5	link/ ln	Create a hard link → file must exist, links and binds the same disk space, if original file is removed, the disk space is still bound to the hard link. Hard links share the same inode.
6	Find the file's character set	file -i → gives the mime type, search for binary, ascii etc.

7	Inode	Metadata (information about other data) for files in the file system held in a flat array. Holds ownership, access modes, and file type. Contents: • Size in bytes • Device ID • User & Group ID • File Mode (i.e. access) • User Flags • Timestamps → file modified, inode modified, and last accessed • Link count • Pointers to disk blocks

File Tools

1	cat	Read a file
2	tac	Read a file backwards
3	Head	Read first few file lines
4	Tail	Read last few file lines
5	read	read from user input → read var → will set the var variable

Find

1. Find a specific file by name find <Starting directory> -name <filename>
2. Finding by type → find <Starting directory> -type <d/f...>
3. Searching depth → find <conditions> -maxdepth <depth>
4. Running a command on all found files → find <conditons> -exec <command> + (the + ends the command, so does \;)
5. Files by last accessed time → -atime <days_ago or -amin min_ago>

    a) a → accessed, m → modified, c → changed

b) use -daystart to count from the start of the current day instead of right now

c) use + for greater than the time, - for less and none for exactly

Finding Stuff

1	Locate (mlocate in suse)	Use updatedb to prepare a database with file locations, then that can be used instead of the slower find
2	which	Shows the full path of (shell) commands (or aliases)
3	whereis	Searches for commands installed and where it is → only for programs no aliases

TAR & ZIP

1	Make a tarball	tar -cpf fileout.tar filename1 filename2..., add p to mantain permissions
2	Extract a tarball	tar -xpf filename.tar (be cautious of 'tarbombs' extract in a directory)
3	tar & gzip	tar -czpf fileout.tar.gz filename1 filename2...
4	Uncompress .tar.gz	tar -xzpf filename.tar.gz
5	Compress to .gz	gzip filename
6	Uncompress .gz	gzip -c filename.gz
7	Compress to .Z	compress filename
8	Uncompress .Z	uncompress filename.Z

1.9 File System

Hierarchy (FHS-V2.3)

Docs: http://www.pathname.com/fhs/pub/fhs-2.3.pdf

1	bin	Essential command binaries
2	boot	Static files of the boot loader → unbootable w/out
3	dev	Device files
4	etc	Host-specific system configuration → must be static, cannot be a binary
5	home (optional)	User home dirs
6	lib	Essential shared libraries and kernel modules →

7	lib<qual> (normally lib64 or lib32, optional)	If multiple library versions are needed like 32 & 64 bit
8	media	Mount point for removeable media → use lsblk to get the names of these
9	mnt	Mount point for mounting a filesystem temporarily
10	opt	Add-on application software packages
11	sbin	Essential system binaries
12	srv	Data for services provided by this system
13	tmp	Temporary files
14	usr	Secondary hierarchy
15	var	Variable data
16	root (optional)	Home dir for root user

Mounting

1	Mounting	mount /dev/<device> destination
2	What disk are mounted	mount
3	Connected disks	lsblk prints out all of the connected devices nicely formatted
4	Mounting on boot	edit /etc/fstab

RAID

1	Name	Redundant array of inexpensive/independant disks
2	Description	Combines mutiple storage devices onto one virtualized disk. Used to improve performance and/or reliability
3	Performance	Improves performance by striping data across disks, allowing simultanious read/write operations of multiple disks.
4	Reliability	Mirrors data on multiple disks to deal w/ disk failure.
5	Levels	RAID has has levels 0,1,0+1,1+0,2,3,4,5,&6
6	RAID 0	Performance → stripes data across multiple disks to speed up R/W
7	RAID 1	Reliability → aka Mirroring, duplicates data to multiple disks
8	RAID 0+1	Reliability w/ Performance → Mirrors of striped data
9	RAID 1+0	Performance w/ Reliability → Stripped mirrors of data

10	RAID 5	Performance w/ some relibility → N-1 disks store data can lose 1 disk
11	RAID 6	Performance w/ Reliability → Like RAID 5 but with N-2 disks. Can lose upto 2 disks
12	Others	RAID 2-4 are rarely used.
13	JBOD	Just a bunch of disks (aka linear RAID), combines several disks into a single logical one.

What would you do to recover a lost the root password to a Unix/Linux system?
Write a locking function in bash
What is a pre-emptive kernel, what does that mean to you?
What is the name and location of the system log on a Unix or Linux system?
find system inof → uname -a
What is the system locale?
Where do the login scripts live? Where would I go to find out how many times a user logged in and from where before their account got locked?
Where are the DNS, Hostname and most other system wide configuration files? How can you edit them?
How could I see if a file system is running out of space. Then: how can you see what is being written that is taking up the most space on that file system?
Apache vs. NGINX
Monitering processes
Why should you never SSH into a production server as root, even if you will be immediately elevating to root?
What's the difference between a named and an unnamed pipe? Is one superior to the other?

2

Networking

2.1 Resources

Books

1. Beginning Linux Programming (3rd) (See the section on sockets)
2. Unix Network Programming
3. Networking for System Administrators
4. Unix & Linux System Administration Handbook, 4^{th}

Links

1. **Network Questions:** https://github.com/kylejohnson/linux-sysadmin-interview-questions/blob/master/test.md

2.2 Connections

Sockets

{FIXME: answer these}
When a client running a web browser connects to a web server, what is the source port of the connection? What is the destination port of the connection?

1	Def	A unix file type with duplex communication
2	Use	Communicating between processes
3	List Sockets	TCP/UDP → Socklist, all → netstat & ss
4	Listening TCP Sockets	netstat -tl

- Attributes: Domain, Type, Protocol

  - Domain → Address family (UNIX → AF_UNIX, TCP/IP → AF_INET, etc)
  - Type → Communication characteristics
    * Stream Sockets (SOCK_STREAM) → Sequenced & reliable 2 way byte stream. Large messages fragmented, transmitted, & reassembled. Order of packets is guarenteed
    * Datagram Sockets (SOCK_DGRAM) → Doesn't establish & maintain a connection. Unsequenced & unreliable.
  - Protocol → UNIX and TCP/IP sockets dont require protocols → use 0 for the default

- Communication Protocols
  1. UDP → AF_INET domain with SOCK_DGRAM connection type
  2. TCP/IP → AF_INET domain with SOCK_STREAM connection type
  3. Others exist, but are less common

Common Sockets

1	Wake-On-Lan	Port 9 → Unofficial
2	FTP	Data transfer: Port 20 Official, Control: Port 21 Official
3	SSH	Port 22 Official
4	Telnet	Port 23 Official
5	SMTP	Port 25 Official
6	WHOIS	Port 43 Official
7	DNS	Port 53 Official
8	HTTP	Port 80 Official
9	SFTP	Port 115 Official
10	HTTPS	Port 443 Official
11	Syslog	Port 514 Official
12	Traceroute	Port 33434 Official

TCP/IP

The application determines which communication protocol is more appro- priate. On the Web, you normally do not want data to go missing during transmission (a piece of text, image, or downloaded software might get lost, with annoying to catastrophic results), hence TCP is the correct choice. For television or voice chat, it is usually preferrable to live with small breaks in the service (a pixellated picture or a brief burst of static) than for everything to grind to a halt while the system arranges for a missing datagram to be

1	IP Packet	A data packet sent by the TCP or UDP protocol. Contains header info and data. 20 header bytes and variable number of data bytes
2	Local host	Means *this computer*, connects to the loopback address → 127.0.0.1 - 127.255.255.254 (IPv4) & ::1 (IPv6)
3	ARP	Address resolution protocol. Maps an address (like IPv4 address) to a device (like a MAC address). Same for IPv6 this is done by NDP (see below)
4	NDP	Neighbor Discovery Protocol, removes necessity of DHCP for configuring hosts, although DHCPv6 does exist
5	MAC Address	Media access control address. Unique identifier assigned to network interfaces for communications at the data link layer of a network segment. Also known as Ethernet hardware address (EHA), hardware address or physical address. MAC addresses are supposedly unique world wide. Find current mac w/ arp
6	Find an IP or site name	dig <site name>/<ip address>
7	Find site info from DNS	whois <site name>
8	DHCP/DHCPv6	Dynamic Host Configuration Protocol. Standard network protocol for IP. Dynamically distributes network configuration parameters, such as IP addresses, for interfaces and services
9	Default Gateway	Path to reach all none local connections. Computer → Def Gateway (usually a router) → ... → destinations router → destination. Use rout to find gw address
10	NAT	Network address translation. Rerouting IP addresses so that there is only 1 internet routable IP for an entire private network. Used synonomously w/ IP masquarading. Used due to IPv4 exhaustion.
11	IPoAC	IP over Avian Carriers. IP packets carried by pigeon. Mike Tyson IT.
12	Subnet mask	Defines locally reachable connections. etc 192.168.178.0/24 means the first 24 bits are masked away and only the last 8 bits are locally reachable. So 192.168.178.0 to 192.168.178.255 can be reached locally
13	CIDR	Classless Inter-Domain Routing, AKA supernetting → removes the necessity for IP classes by masking IP bits by necessity

14	Packet filter/firewall	Filtering based on origin from various IPs

IPs (ranges/subnets) reserved for private use/"non-routable" (RFC 1918)?

IP Class	From	To	CIDR
Class A	10.0.0.0	10.255.255.255	10.0.0.0/8
Class B	172.16.0.0	172.31.255.255	172.16.0.0/12
Class C	192.168.0.0	192.168.255.255	192.168.0.0/16

How does a switch get a mac address?

What type of packet to discover a router?

A TCP connection on a network can be uniquely defined by 4 things. What are those things?

Internet

1	HTTP/HTTPS	Hyper text transfer protocol / secure. Request - response protocl for server-client computing.
2	SMTP	Secure messaging transfer protocol
3	DNS	Domain name service, look up IP addresses from human readable names. Use whois or dig as a cmd line tool.

Tools

1	ifconfig	Network configuration & querying the setup of a network interface
2	ip	newer version of ifconfig, use ip addr show to list all connections
3	whois	Look up info in DNS about site
4	arp	Look at the computers hooked up in the subnet and the hardware addresses known
5	route	show / manipulate the IP routing table
6	traceroute	print the route packets trace to network host
7	Ping	Uses the control protocol, ICMP, see if communication is possible. Use ping6 to test IPv6 connections

8	LDAP	Lightweight directory access protocol. A lightweight database for storing various bits of info. Common attributes: • **dn** → distinguised name: Search path ex. dn: uid=simon,ou=people,dc=navy,dc=mil • **o** → organization: Often the top level entry • **ou** → organization unit: logical subdivision • **cn** → common name: most natural name to repr entry • **dc** → domain component: used when the model is based on DNS • **objectClass** → Object class: Schema used for this entry

2.3 Remote Connections

HTTP Servers

1	Purpose	HTTP daemons, can handle multiple websites, requests, manages load on a server etc.
2	Apache	Apache License, most popular and flexible
3	Nginix	BSD License, 4x requests per second and less memory than apache but is less flexible.
4	Varnish	FreeBSD License, heavily multithreaded making it good for content heavy dynamic webpages

SSH

1	Encryption	All communiucations are encryted → handshake determines the encryption protocol and proime number, they then share the public keys and keep a secret key

2	Keys	Secret & public key. Put public key on sever, server sends message to client, client uses secret key to send a return message which confirms the connection.
3	Generating keys	ssh-keygen -t dsa
4	X forwarding	-X (unencrypted), -Y (encrypted)
5	File transfer	SFTP/SCP are the ssh tunnel file transfers, sftp being the upgraded version of scp.
6	SSH Hardening	1. Disable SSH protocol 1 2. Reduce the grace time (time to login) 3. Use TCP wrappers (always good to check) 4. Increase key strength (maybe go to 2048-bit keys) 5. Check the defaults and disable a few options
7	Server	rcsshd. To start the ssh server run: rcsshd restart

TLS/SSL

1	TLS	Transport Layer Security
2	SSL	Secure Sockets Layer
3	Encryption	By key pairing
4	Digital certificates	relies on a set of trusted third-party certificate authorities to establish the authenticity of certificates. Ensures that the public key holder is who they claim to be (perventing man in the middle attacks)
5	File transfer	FTPS → FTP SSL or HTTPS → HTTP SSL (or secure, etc)

FTP & Telnet

1	FTP	File transfer protocol. Often used with SSL liscences for FTPS
2	Telnet	Provides cmd line access to a remote host like ssh. Security concerns has made ssh the prefered communication method

Mail Servers

1	SMTP	Secure mail transfer protocol
2	MX record	Mail exchange message

OSI

ISO OSI reference model → Open Systems Interconnection model. 7 layers each of which only see 1 up and 1 down. So the network doesn't care if its copper or fiber, the application doesnt care if its IPv4 or Appletalk, each layer is supposed to be independant of the others.

	OSI Model			
	OSI Layer	**Protocol Data Unit (PDU)**	**Function**	**TCP/IP Example**
Host Layers	7) Application	Data	High level APIs including resource sharing and remote file access	SSH, FTP, ...
	6) Presentation	Data	Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption	Character conversion, data formatting, MIME
	5) Session	Data	Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes	Network Socket
	4) Transport	Segment (TCP) Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing	TCP or UDP
Media Layers	3) Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control	IPv4/v6
	2) Data Link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer	Ethernet/wifi adapter
	1) Physical	Bit	Transmission and reception of raw bit streams over a physical medium	Fiber optic, Radio

DNS

DNS Record Types: https://en.wikipedia.org/wiki/List_of_DNS_record_types

1	'A' record	Address record → Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc.
2	CNAME record	Canonical name record → Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.
3	'NS' record	Name server record → Delegates a DNS zone to use the given authoritative name servers
4	'PTR' record	Pointer record → Pointer to a canonical name. Unlike a CNAME, DNS processing stops and just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD.
5	DNS forwarder	specific DNS requests are forwarded to a designated DNS server for resolution
6	Reverse Lookup	Double check an IP address by looking up the DN based on the IP

Switches

1	Use	Connects a network to a series of connections using packet switching, only sends data to the target computer.
2	OSI Layer	Layer 2/3 → used hardware addresses (MAC) to switch packages. Some include network capability using IP addresses
3	Hubs	A hub sends out all packages that come in to all attached connections. A switch selectively sends packets to the correct hardware address.

Routing

1	OSI Layer	Network layer 3

2	Schemes	• Unicast → Sent to single node • Broadcast → Sent to all nodes • Multicast → Sent to a group of nodes • Anycast → Sent to anyone of a group of nodes • Geocast → Sent to a geographic area

Terms

1	Proxy	A server that acts as an intermediary for requests from clients seeking resources from other servers.
2	IPS	Internet Provider Security → aka registrar tag, used by domain registrar to administer a domain name registration service and related Domain Name System (DNS) services
3	DOS	Denial of service → overloading the bandwidth of a server to take it offline

Servers

Server Automation: http://www.infoworld.com/article/2609482/data-center/data-center-revie
html

1	Server Automation	Puppet, Chef, Ansible, and Salt make it much easier to configure and multiple identical servers.
2	Puppet	Ruby based server automation
3	Chef	Ruby based server automation, similar to puppet
4	Ansible	Python based server automation,
5	Salt	Python based server automation, strength: scalability and resiliency

Network Configuration

1	LLDP	Link Layer Discovery Protocol. Automatic network configuration mapping. Open source
2	CDP	Cisco Discovery Protocol. Proprietary
3	SNMP	Simple Network Mangement Protocol.
4	LCP	Link Control Protocol, part of PPP (see below)

Point to Point Protocol (PPP)

Definition RFC: https://tools.ietf.org/html/rfc1661
PPP Over AAL5 (PPoA): https://tools.ietf.org/html/rfc2364
PPP Over Ethernet (PPPoE): https://tools.ietf.org/html/rfc2516

1	Definition	
2	Benifits	
3	PPPoE	PPP over ethernet.
4	PPPoA	PPP over AAL5.

VLAN

VPN v. VLAN: http://www.differencebetween.net/technology/difference-between-vlan-and-vp
Subcategory of VPN, purely layer 2 construct making computers connected in various ways act like they are all connected on the same switch. Allows for breaking up networks that are on the same switch and combining ones that arent.
Summary: 1. VPN is a method of creating a smaller sub network on top of an existing bigger network while VLAN is a subcategory of VPN 2. A VLAN is used to group computers that are not usually within same geography into the same broadcast domain while VPN is most commonly related to remote access to a companys network

Ethernet

1	Connector	RJ-45
2	Wire	UTP $\rightarrow$ unshielded twisted pair (copper)

3

Programming

3.1 GIT

Setup

1	Get a repo	git clone
2	Make a repo	git init
3	Pull an existing repo	Use init or clone the repo then pull
4	Remote repos	git remote → lists the remote repos, git remote add "name" "url"
5	Configuration	git config → complicated, but add email and user with git config –global user.email & user.name

3.2 Terms

Programming

1. **Agile:** Software development strategy. Values:
   a) **Individuals and Interactions** over processes and tools
      i. Pair programming → 1 station 2 programmers, driver & navigator/observer
      ii. Colocation → Team members in the same area
   b) **Working Software** over comprehensive documentation
   c) **Customer Collaboration** over contract negotiation
   d) **Responding to Change** over following a plan

3.3 Python

Standard Library Scripting

Python Standard Library: https://docs.python.org/3/library/index.html

1	OS Module	Miscellaneous operating system interfaces. Some attributes are cross platform, some platform specific. OS.path contains all the file name manipulation tools.
2	Subprocess Module	Meant to replace parts of the OS module. Run subprocesses, use pipes, etc.
3	Sys Module	Use for passing simple arguments (use argparse for more complicated argument passing). Get system information and shell variable analogs for the python environment.
4	Argparse	A more plush way of getting command line arguments. Auto-generates a help screen.
5	Shutil Module	Contains file and directory functions

Package Development

Pydocs: https://docs.python.org/2/tutorial/modules.html#packages **PCU:** https://pythonconquerstheuniverse.wordpress.com/2009/10/15/python-packages/

1	Making a Package	1. Contains a set of modules and atleast one __init__.py 2. Append the location of the module to PYTHONPATH, the working directory is checked last → sys.path.append(<Package Location>) 3. The __init__.py module is run at the start, so to have all submodules nicely loaded use from add import add to be able to call <Package Name>.add instead of <Package Name>.add.add

Environment

1	Virtualenv	Isolated working copy of Python allowing the altering of a python setup without affecting other projects
2	Use	Packages installed here will not affect the global Python installation.
3	New virtualenv	In a clean directory run: virtualenv <Dir Name>, add –no-site-packages to not use already installed packages
4	Add packages	Call pip from the correct env directory. This will install the package in the virtual environment directory instead of the main installation
5	Activate	From <Dir Name> use: source activate.
6	Deactivate	Call: deactivate

3.4 MySQL

Relational Database

Wiki: https://en.wikipedia.org/wiki/Relational_database

1	SQL	Structured Query Language
2	Relational Model	Table rows have unique keys. This allows for columns of 1 table to be linked to columns in another on some shared attribute
3	Databases	1. MySQL 2. Mariadb 3. sqllite 4. Postgres 5. couchdb

Start mysql server: rcmysql start