Linux Mastery : Master the linux Command Line

-Ziyad Yehia

Ubantu.

Command line terminal → Ctrl + Alt + T >> Open

Ctrl + D >> close

Command $ echo Hello

→ Hello

* clear or ctrl + L

* exit or ctrl + D

Command $Cal

→ Calender of this month.

$ Cal 2017

→ get calender of this year

$ Cal -y

→ get Calender of this Year.

Command Name -Options Input

Command store in shell Operand

ex1: Cal option -y 2017

ex2: Cal -A 1 option -B 1 12 2017

Command Name ~~Command~~ ~~Command~~ multiple Input

Command Input or ~~operand~~ ~~Command Input~~ Command I/P

Command $ clear or Ctrl + L

Command $ history

→ list of all previous Command

Short Command : -A

Long Command --After

--before

$!8

→ execute 8 number Command → Run particular Command from history

$!!

→ executed previous Command

→ $ history -c ; ~~history -w~~

→ clear all previous Command.

example: Gift

Command ⇒ Shell ⇒ Meaning

(Interpreter)

Terminal = Window into shell

Commands are just Text you type in terminal - Commands are interpreted by Shell - Different Shell can interpret the same text in different ways. - The terminal is the window to the Shell. {Terminal middle piece bet^n Command & Shell}

Command are Case - Sensitive. - Command Names need to be on Shell Search path - CommandName found from Left to Right.

$echo $PATH

→ Provide Shell add^r - where Commands store - whenever we type Command interpreter look into this file if he found he execute task.

$ which Command-name

→ Provide Pathwere Command store

Page No.
Date

Linux Manual

Help to know command in Depth

Manual Structure

Section	Contains
1	* User Commands {Regular User? No sudo need}
2	System calls
3	C Library Functions {GUI, coding}
4	Devices & Special Files {CD drive, USB Port, device}
5	* File Formats & Conventions. {file & folder}
6	Games {in build game}
7	Miscellaneous
8	* System Administration {sudo previlage}

$ man -k which

→ lcf (1) - Determine which of the historical version of Config.

which (1) - locate a Command

sol(6) - a Collection of card games which are easy to play at

show all option

man file Section

To get detail of manual.

$ man 6 sol

→ get detail manual of Section 6.

Synopsis :- which [-a] filename... → Command Name.
→ options. → more than one input

[ ] optional to use

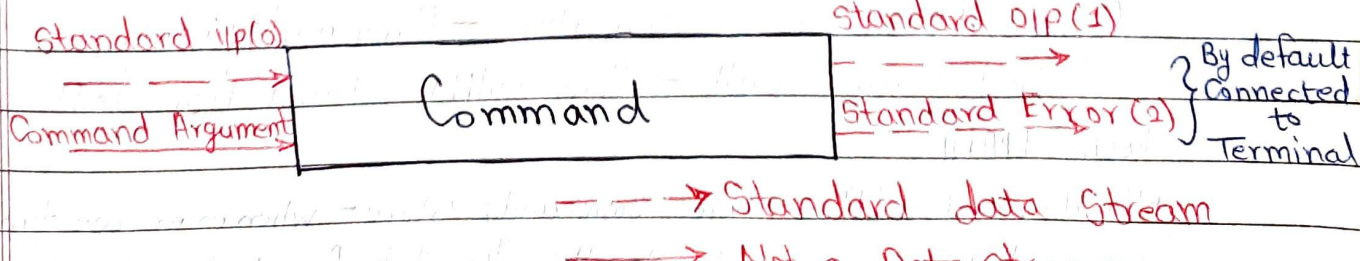
< > have to have it

| pick one or other, Couldn't have both.

$ help cd

→ get manual of some Command which not get by man Command

Command Input + Output



IMP

Standard Input = 0, Standard o/p = 1, Standard Error = 2.
> will overwrite a file before writing to it.
>> will append to what's already there.

Page No.
Date

Command Argument i/p command ot o/p directly connected to keyboard or read from standard i/p. example :- Cat Command.

Note :- Standard o/p or error can be redirected to any other Command {Concept of Pipeline}

* Redirecting

$ Cat 1> output.txt ----- Content add into Output.txt file

1 indicate standard o/p

Truncation → Linux execute last updated Command & over write the Content previously appear.

To avoid Truncation

$ Cat 1>> output.txt. ------ Apending to file.

Redirecting Standard ~~o/p~~ error (2)

$ Cat 2> error.txt

example :- $ Cat date xyz 2> error.txt

Combine Redirection.

$ Cat 1>> message.txt 2>> error.txt ---- when error occure mess write into error.txt otherwise in message.txt.

Take i/p from txt file and display on Terminal.

$ Cat 1> Message.txt

Hello World !! enter data into message.txt

^C

$ Cat 0< Message.txt. --- cat Command receive i/p from message and show o/p (by default) on Screen.

Redirect from one file into another.

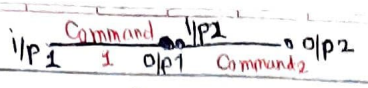
$ Cat 0< ~~input~~ message.txt 1> new-file.txt

i/p from message.txt — o/p to new-file.txt.

i/p from standard i/p [keyboard] → [Cat Command] → o/p to standard o/p [display]

Piping Connects STDOUT of one Command to to STDIN of another Command.

Page No.
Date

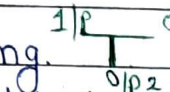
Piping 

Piping Connects STDOUT of one Standard o/p of Command 1 given as input for ~~Std~~ Command 2.

$ date | cut --delimiter=" " --fields=1 1> today.txt

Command 1 | Piping ↓ cut ↓ Command 2 Separate o/p with Space bar. Select 1 field Store o/p into .txt file.

o/p → today.txt have Mon into it. - Can have multiple pipe.

Tee Piping. 

$ date 1>date.txt | cut --delimiter=" " --fields=1 1> today.txt.

above Command Not work because o/p of date Command store into date.txt so next Command not work. on o/p of date.

To resolve this we have to use Tee Command.
Redirection of STDOUT breaks Pipeline.

$ date | tee date.txt | cut --delimiter=" " --fields=1

once you redirect file (or save file >) you ~~can~~

~~Note~~: Now onward you cannot use pipe lining. - but to extend use | tee to extend. — To save a data "Snapshot" without breaking Pipelines, use the Tee Command.

Xargs

xargs allows ~~Command~~ Convert Pipes stand. output data into Command line argument — Some Command only accept Command line argument only. example: echo - echo Command only allow Command line arg

$ date | xargs echo

If a Command doesn't accept STDIN, but you want to pipe to it, use xargs.

Page No.
Date

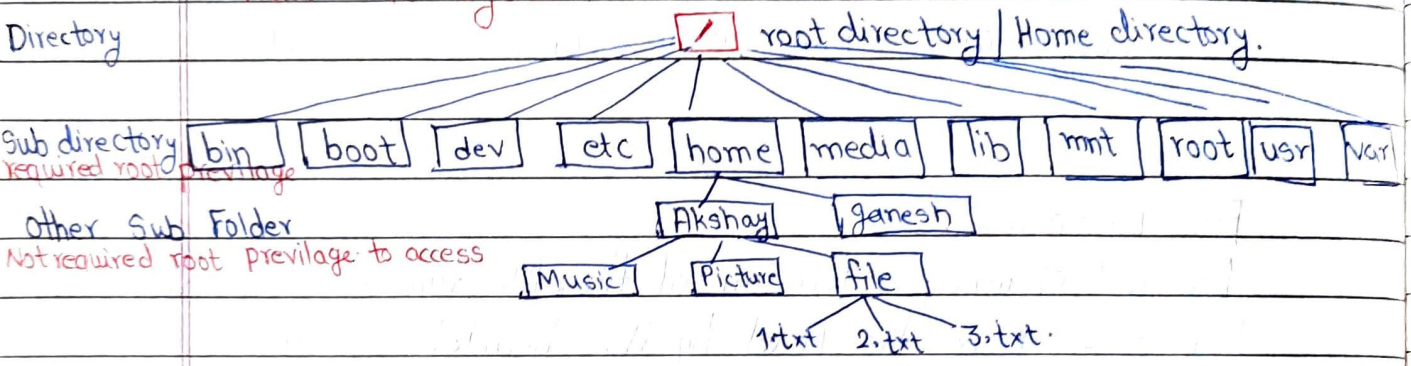
Aliases. {alias aliasName = " Command "}

easy to remember Nick name.

1) Create Note (.txt) file with name .bash_aliases
2) Write any Command in file. with alias aliasCommand = 'Command'
   alias getdates = 'date |tee fulldate.txt | cut --delimiter = " " --fields = 1 |tee shortdate.txt |xargs echo'
3) Reboot Terminal

Create your own Command.

Linux File System.



/ The very top (Root) of the file Tree. Hold everything.
/bin Store Common linux user Command binaries e.g. date, cat
/boot Bootable linux kernel and bootloader Config. file.
/dev File representing devices. tty = terminal fd = floppydisk
/etc Administrative Configuration files
/home Home directory for regular users are stored.
/media removable media (USB, Sticks, External HDD)
/lib Contains Shared libraries needed by applications in /bin and /sbin to boot the System.
/mnt A place to mount external devices. ~~This can still be~~
/misc a directory used to Sometimes automount filesyst on request.
/opt Directory to store additional s/w.
/proc Info. about System resources
/root The Superuser
/sbin Contain administrative Commands (binaries) for the root (Super) user.
/tmp Contains temporary files used by running apps
/usr Contains files pertaining to users that in theory don't
changes after installation.

Page No.
Date

Navigating File System

pwd → Print Working Directory
ls → list out files in working directory.
~$cd downloads → Relative path
~$ cd /home/Akshay/downloads → Absolute Path - Start at base directory.
cd .. → Back one folder.
cd → Home folder move

(Current User Home directory)

Note :- Type initial word of directory or file & then press Tab to create absolute path.
Go through GUI - Right Click - Open in Terminal

File Extensions

By just changing extension of file, file type does not change in linux - In linux file Read by it's header & type define by the same.
$ file 123.txt

Note :- When you put random extension to file then linux open that file effectively but when you give known extension then it Produce error.

example 123.txt (original file) 123.pdf (error file) 123.akshay (Read like txt file)

Wild cards *

$ ls *.txt
display all data from all directory with extension of .txt.

$ ls ?.txt
? match only one place example A.txt, B.txt, ABCD.txt ✗

$ ls file[1234567890].txt or $ ls file[0-9].txt.
[ ] match with define character only example file1, file2, file3 ✓
fileA fileB fileC ✗

Anything that matches the pattern will be passed as a Command line Argument to a Command.

* Matches anything, regardless of length.
? Matches anything, but just for one Place.
[ ] Matches just one place, but allows you to Specify Options.

Creating Files & Directories.

$ touch file.txt ---- Create file.
$ touch ~/Documents/file.txt ---- Create file in Documents.
$ echo "Hello" > hello.txt ---- Write Result into hello.txt. Redirecting.

$ mkdir Akshay ---- Create directory.
$ mkdir -p Akshay/Ashok/Dange --- Create directories with path. (Create unavailable Path.)
$ mkdir Akshay Dange ---Create two Separate directory.
$ mkdir "Akshay Dange" ----Create Single folder with ~~Happy~~ Akshay D
Directory name are case sensetive akshay Akshay AKSHAY can be exist

$ mkdir {jan, feb, March}-{2010, 2011} Brace Expansion Method
create folder jan-2010, jan-2011, feb-2010, feb-2011, March-2010, March-2011

$ mkdir {jan, feb, March}-{2010..2020}
$ mkdir {jan, feb, March}-{2010, 2011}/file{1..100}
Create file 1 to file 100 in each of the folder.

$ touch file{A, B, C, D}.txt. --- Create fileA.txt, fileB.txt, fileC.txt, fileD.txt
$ touch file{A-D}.txt ----

Delet File & Directories.

$ rm file.txt file1.txt Documents/file.txt
file Absolute path
$ rm -r *.txt ---delete all txt file Recorsively. (multiple file)
$ rmdir Akshay Documents/Akshay ---- delete only empty folder
$ rm -ri *.txt ---- interactive file deleting.

Copying Files & Directories

$ cp file1.txt file2.txt ---- Copy file1 to file2.
(Source, destination)

$ cp file1.txt file2.txt Document/ --- Copy file 1 & 2 to Docu
(Source; Last is destination)

$ cp -r Akshay Dange --- Copy Akshay directory to Dange direc

Moving + Renaming.

$ mv oldname.txt newname.txt → Renaming

$ mv newfolder/* Desktop → move folder Containt to Desktop
(source, destination)

Editing files.

$ nano file.txt.

^O write out → Save file.

^X Exit

^R Read file → Read data from another file

^W Where Is → Search - less case sensitive.

M-C → Alt + C → Case Sensitive Search.
(M- = Alt, Esc or Cmd)

^\ → Replace

^K → Cut

^U → Paste

Locate Command {List only files} locate data base file.

$ locate *.txt ---> locate all files with .txt extension

$ locate -i *.txt ---> case insensitive.

$ locate -i --limit 3 *.txt ---> locate first 3 files only

$ locate -S ----> give database Size

$ locate --existing *.Confi ---> give files which actualy exist.

To access file need to update database

$ updatedb ---> But need super user access.

$ Sudo updatedb

Note :- Once you add any file then update database so locate Command work effectively.

either you be on that Folder or in subfolder

Page No.
Date

Find Command

list out all files & directory.

Note: Find Command doesnot need data base in order to work.

$ find /home/Akshay/Documents.

$ ~/Document/ $ find ---> list out all files in Document.

$ find . -maxdepth 1 ---> list out content upto maximum dept of 1 Position.

$ find . -type f ---> list out only file type

$ find. -type d ---> list out only directory type

$ find . -name "5.txt" ---> find file by name of file

$ find . -iname "5.txt" ---> find name with less Case Sensitive

$ find -type f +size +100k ---> Search file more than 100kByte

$ find . -type f -size -100k ---> Search file less than 100kByte

$ find / -type f -size +100k -size -5M ---> 100k < f < 5MB.

$ find / -type f -size +100k | wc -l ---> give only list of Count.

$ find / -type f -size +100k -size -50k -exec cp {} ~/Desktop/Copy-here/ \;

(execute) -ok --- same as exec only ask permission

---> Copy file Size more 100k & and less than 50kB to Copy-here fold

Create 500 folder & 100 files in each folder - Create one file into it & move out on desktop folder randomly.

$ mkdir Akshay

$ mkdir /Akshay/folder{0...500}

$ touch /Akshay/folder{0...500}/{0..100}

$ touch Akshay/folder$(shuf -i 1-500 -n 1)/needle.txt ---> Create needle.txt in random folder.

$ find Akshay/ -type f -name "needle.txt"

$ find Akshay/ -type f -name "needle.txt" -exec mv {} ~/Desktop \;

Viewing File

$ Cat file1.txt ---> Read file on cmd.

$ Cat file1.txt file2.txt file3.txt > Beautiful.txt ---> Combine all files

$ tac file1.txt ---> Reverse the lines in file / last line 1 & 1 line last

Reverse Vertical Position of line

$ rev file1.txt ---> Reverse file Containt Horizontally Last word first & first word last

$ less file1.txt ---> good Representation of file Q to move out
$ head -n 2 file.txt ---> Provide Specific Number of line | bydefault 10
$ tail -n 5 file.txt ---> Shown last 5 line

Sorting Data

$ Sort word.txt ---> Sort alphabatically words.
$ sort -r word.txt ---> Reverse sorted output.
$ sort numbers.txt ---> Not Sort by number value instead alphabatically Sorted.
$ sort -n numbers.txt ---> sort by hole number. (numerically)
$ sort -u number.txt ---> delet Duplicate number only show unique number
$ ls -l /etc | head -n 20 | sort -k 5n ---> Sort data of Coloum number 5. by numeric size
$ ls -l /etc | head -n 20 | sort -k 6M ---> Sort file by month.
-k is keydef 3nr means sort using Column 3, -n & -r options

Search file using Grep Command

$ grep e hello.txt ---> Find letter e in hello.txt file. - Case sen
$ grep -c e hello.txt ---> get count of result
$ grep -i e hello.txt ---> less case sesitive
$ grep -i "Hello World" hello.txt ---> Phrase Searching.
$ grep -v e hello.txt ---> Reverse Search - finding lines who don't have e in it.
$ grep -ic e hello.txt Hi.txt ---> Select multiple file to search e

File Archiving & Compression

Step 1 1) Create tar file (Combine all required file in single folder)

Step 2 2) Compress folder.

① $ tar -cvf ourarchieve.tar file[1-3].txt ---> c for Create
Great view ~~folder~~ files v for interaction & f for folder

$ tar -xvf ourarchieve.tar ----> extract files from .tar folder
~~unfolder~~ unfold

Compression

gzip	bzip2
1) More Faster	1) Slower
2) Has ~~more~~ less Compression power	2) Has more compression ab

② bzip2

② $① gzip ourarchieve.tar ---> Compress

$① gunzip ourarchieve.tar.gz ---> Uncompress folder
② bunzip2 bz~~ip~~2

$ zip ourthing.zip file1.txt file2.txt file3.txt.

$ unzip ourthing.zip

Combine Operation ~~to~~ Archiving + Compression.

$ tar -cvfz ourarchive.tar.gz file[1-3].txt. ---> Compress file.

$ tar -cjvf ourarchive.tar.bz~~ip~~2 file [1-3].txt ---> Compress into bzip2 format

$ tar -xvfz ourarchive.tar.gz } extract file

$ tar -xvfj ourarchive.tar.bz2

#! /bin/bash/Python 3
Shebang — Path from where this file execute as.

Automation the Workflow

Bash Script

$ nano our-script.sh ---> Create Bash file

In our-script.sh file

#!/bin/bash ----> tell this is bash file interpret as such.
called shibang line - below all interpret as Commands

to get path type $which bash on cmd

```
echo "Hello World!"
```

$ bash our-script.sh ---> Run bash file

```
Hello World!
```

Task Create bash file with creating one folder on desktop with 100 file in it and Crea put info all folder in one file.

```
#! /bin/bash
mkdir ~/Desktop/magic
cd ~/Desktop/magic } changing Path is important.
touch file{1..100}.txt
ls -lh ~/Desktop/magic > ~/Desktop/magic.log
```

$ bash our-script.sh ---> execute file

Create Backup of Desktop folder.

$ nano backup.sh

```
#! /bin/bash
tar -cvfz backup.tar.gz ~/Documents
```

$ bash backup.sh

Create own Command with Bash Script.

Alias also perform same → But for multiline Script you need .sh help

1) Create bin directory in Home ~

~$ mkdir bin

2) $ mv backup.sh ~/bin ---> Move file to ~~back~~ bin

$ mv backup.sh backup ---> Rename file

3) $ chmod +x backup ----> give execute permission to backup file so act as Command line

4) $ backup ---> Perceive error need to add path

5) $ nano .bashrc

go to last line of file and add Path - because file doesn't have path & updated shell

PATH = "$PATH:$HOME/bin" ---> bin folder path added into shell Path.

$ backup ---> Create backup Now onward put all .sh file into bin to run as sh Command

Scheduling using Cron

automate your workflow by scheduling your scripts to run when you want.

Cron allow to execute any ~~tim~~ Command at a given Specific time.

$ Crontab -e ---> edit Cron file

Choose [1-3] : 1

Cron Command line divide into 6 Column & 1 Row.

min	hour	day of month dom	month mon	day of week dow	Command
[1-59]	[0-23]	[0-31]	[1-12]	[SUN - SAT]	
30	14	11	11	SUN	

2:30 PM on 11 day of month on nov on Sunday

* any value we don't Care. Space Separate Command

* * * * * ~~ba~~ echo "Hello World!" >> ~/Desktop/hello.txt

Run Command on every min and add hello world on hello.txt file

0,15,30,45 ---> Run at every 15 min

*/7 ---> Run at every 7 min.

Create backup On every FRI at 23:59 with bash Command

59 23 * * FRI bash ~/bin/backup

modify something in backup file like,

$ mkdir ~~backups~~ ~/Backups.

$ nano backup.sh

#!/bin/bash

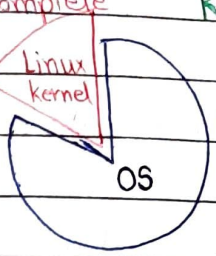
tar -cvfz ~/Backups/backup.tar.gz ~/{Documents, movie, Pictures}

Date >> ~/Backups/date-log.txt.

Operating Software

Page No.
Date

Linux Kernel Complete
GNU Project
"Linus Torvalds"



Kernel Provide Slw to access resources of Computer like memory, CPU, I/O Port - Communicate bet^n Hlw & Slw

GNU Project "Richard Stallman"

Provide Open Source (Free) Slw - Open Source Code - open access to all file & System.

4 Freedoms

1) The freedom to run program as you wish.
2) The freedom to study how the program works & change it so it does your Computing as you wish.
3) The freedom to Redistribute Copies so you can help your neigh
4) The freedom to modify / develop new version - can give the whole Community a chance to benefit from your change.

Download Source Code of linux & Update Code as per Requirement.

gnu.org
Software → GNU OS.
Coreutils package download. → Stable → bottom of list → 8.28.tar.xz

Access to Source Code.

Cmd

$ tar -xJf Coreutils-8.28.tar.xz ——→ Coreutils Hold all Command
$ ls Coreutils-8.28.
$ cd src
$ ls | less
$ ls | grep ls
ls.c → open
$ nano ls.c
open source code of ls. -- 1443 line
go to main code
{ printf ("Hello there you beautiful People \n"); modify ls Command
int i;
Save file

To compile this .c file need Compiler.

/src $ sudo apt-get install gcc ---- get C Compiler
go back cd..
Coreutils-8.28 $ ls } at main folder we found Compiler.

Update: ① download - unzip - go to source folder
② ls.c which modify.
③ download compiler gcc (gnu c compiler)
④ Configure Compiler with new architecture
⑤ install make command

Page No.
Date

$ bash Configure ——→ Great Appropriat modification in Configure gcc Compiler

8.28 $ sudo apt-get install make

8.28 $ make ---> Compiling machine code to binary code

$ sudo make install --> install all new packages which we download

Reopen Terminal

$ ls

get output

To remove Command which edited.

src $ nano ls.c 1443 line

remove specific line

8.28 $ make && sudo make install.

Logis
Media
Textile
Electric

Reopen

$ls

700
400
2000
200

Software Repositories.

Big library of s/w

help.ubuntu.com/community/Repositories/Ubuntu.

packages.ubuntu.com

On Cmd.

$ lsb_release -a ---> Ubuntu distribution Version - Packages

Type of Repositories.

1) Main - all free & Open Source s/w - maintain by ubantu
2) Universe - Community - maintained free & open-source s/w
3) Restricted - Proprietary drivers for devices.
4) Multiverse - s/w restricted by Copyright or legal issue

A s/w Repository is like a library of s/w.

Each Package has other related Packages.

Required - Mandatory for the Package to fun"

Recommended - Needed for normal use

Suggested - Suggested for Interest

Enhances - Enhances the Package in some way

apt - advance packaging tool

Search Packages for docx files.

$ apt-cache Search docx ---> Search in cache memory
$ apt-cache Search docx | grep text
get package Name with detail.
$ apt-cache show docx2txt ---> get info. about Package.

Cache memory hold at /var/lib/apt/lists

Update & Upgrade cache Package

$ sudo apt-get update ---> get updated list from Repository.
Cache = Repository. {enable internet}
$ sudo apt-get upgrade ---> upgrade all s/w.

Install New Software

~~$ sudo apt-search~~
$ apt-cache Search xeyes search for s/w or Package.
get package name with description.
$ apt-cache show x11-apps
give info. about package
$ sudo apt-get install x11-apps download x11-apps.
$ xeyes ---> Run application
$ man xeyes ---> manual to know more ability.

$ xman ---> graphical window for manual.

Downloading Source Code

$ sudo nano /etc/apt/Sources.list
remove # ahead of deb-src lines to activate Source Code.
Save file.
$ sudo apt-get update ---> update
$ sudo apt-get install dpkg-dev → s/w to unpack Source code.
$ sudo apt-get Source x11-apps.

enter into directories
look for ~~xe~~ xeyes.c file
$ nano xeyes.c

Uninstalling Packages.

$ sudo apt-get remove x11-apps --> Configuration file still Preser in system for future use.

$ sudo. apt-get purge x11-apps --> Remove all info.

$ sudo apt-get autoremove ---> remove all other dependency Packages

$ sudo apt-get clean ---> Remove ~~all~~ archieve/compress files. which placed at Var/cache/apt/archives

$ sudo apt-get autoclean ---> Remove unused archieve.

www. virtualbox.org/Downloads
windows hosts → install → uncheck 3 shortcut. → Finish

www.ubantu.com/download/desktop
ubantu updated Version - Download - Not donate - SAVE

Virtual Box - New - Ubuntu - Linux - Ubantu (64)
Allocate green zone memory - Create Virtual Hard disk - dynamically 20 GB (default) - Create
Storage - Control IDE - load disk - ubantu 32 image - ok