

မာတိကာ

စဉ်	အကြောင်းအရာ	စာမျက်နှာ
၁။	Linux ပိတ်ဆက်	1~3
၂။	CentOS Linux	4
၃။	Installation centos linux on virtual box	5
၄။	user, application, kernel and shell	6~8
၅။	linux machine နဲ့ Terminal ဖွံ့ခြင်း	9
၆။	linux file system hierarchy	10~12
၇။	linux ခဲ့ command syntax အကြောင်း	13~15
၈။	file paths on linux system	16
၉။	linux စက်ပေါ်တွင် directory တစ်ခုမှတစ်ခုထိ ကူးပြောင်းခြင်း	17
၁၀။	file နှင့် directory များကို list ထုတ်ကြည့်ခြင်း	18~19
၁၁။	file နှင့် သက်ဆိုင်သော	20~21
၁၂။	directory နှင့် သက်ဆိုင်သော	22~23
၁၃။	date and time	24~26
၁၄။	copy ကူးကြမယ်	27~28
၁၅။	move and rename files and directories	29~30
၁၆။	command တစ်ခုနှင့် ဆိုင်သော အကူအညီများကို ကြည့်ရေးခြင်း	31~32
၁၇။	head, tail, cat, more, less	33~34
၁၈။	creating and editing text file in linux	35~36
၁၉။	creating user account	37~39
၂၀။	creating group	40~41
၂၁။	user account နှင့် သက်ဆိုင်သော	41~43
၂၂။	group နှင့် သက်ဆိုင်သော	44
၂၃။	superuser Vs normal user Vs administrator	45~46
၂၄။	file နှင့် directory/folder များကို permission သတ်မှတ်ခြင်း	47~52
၂၅။	How to prepare for RHCSA Exam	53~54
၂၆။	Linux Shell Scripting	55~56
၂၇။	Special Permission	57~58
၂၈။	Access Control List (ACL)	58~61
၂၉။	default permission and umask	62~64
၃၀။	compression and decompression	65~68
၃၁။	password management	68~71
၃၂။	RPM (redhat package manager)	71~72
၃၃။	YUM (yellow-dog update & modify)	73~75
၃၄။	RPM Vs YUM	75
၃၅။	adding repo link	76~77

၃၆။	controlling services (Services များကို စီမံခြင်း)	78~80
၃၇။	(ssh အပိုင်း) ssh ဖြင့် password ကို အခြေပြု၍ remote login ဝင်ခြင်း	81~82
၃၈။	(ssh အပိုင်း) ssh with private-public key	82~83
၃၉။	(ssh အပိုင်း၃) root account ဆီသို့ ssh ဖြင့် remote login ဝင်ခြင်းမှ တားဆီးခြင်း	83~84
၄၀။	(ssh အပိုင်း၄) normal account အား ssh ဖြင့် remote ဝင်ခြင်းမှ တားဆီးခြင်း	84~85
၄၁။	(ssh အပိုင်း၅) ssh ၏ default port number အား ပြောင်းလဲခြင်း	85~86
၄၂။	rsylog	87~89
၄၃။	system-journal	89~91
၄၄။	The True Happiness is Playing with linux	91~92
၄၅။	network configuration with nmcli	92~94
၄၆။	network configuration (manually)	95~96
၄၇။	hostname	97
၄၈။	root password recovery (redhat, centos, fedora)	98
၄၉။	scp (secure copy)	99
၅၀။	rsync ဖြင့် ကူးယူခြင်း	100
၅၁။	sftp (secure file transfer protocol)	101
၅၂။	identifying file system and device	102~104
၅၃။	how to set time with timedatectl command	105~106
၅၄။	linking, hard link, soft link	106~107
၅၅။	finding via linux terminal	108~112
၅၆။	controlling jobs	112~113
၅၇။	controlling processes	114~115
၅၈။	monitoring the system with top	115~116
၅၉။	grep ဖြင့် စာသားများကို ဖွဲ့ထုတ်ခြင်း	117~119
၆၀။	at ကိုအသုံးပြု၍ job တစ်ခုကို အချိန်ယေားဖြင့် run စေခြင်း	119~120
၆၁။	crontab ကိုအသုံးပြု၍ job တစ်ခုကို အချိန်ယေားဖြင့် ပုံမှန် run စေခြင်း	121~122
၆၂။	kernel update	122~123
၆၃။	System cron jobs	123~124
၆၄။	SELinux (Security Enhanced Linux)	124~126
၆၅။	HD (hard disk) အား partition ပိုင်းခြင်း format ချုခြင်း mount လုပ်ခြင်း	126~128
၆၆။	virtual memory (swap)	128~129
၆၇။	linux firewall (iptables)	130~135
၆၈။	LVM – Logical Volume Management	135~137
၆၉။	Accessing Network Storage (NFS)	138~139
၇၀။	Accessing SMB (SAMBA) File System	140~142
၇၁။	Firewalld (Linux New Generation Firewall)	143~144

Bonus Track

၂၂။	Network Bonding in redhat, centos, fedora linux server	145~148
၂၃။	UFW : Uncomplicated Fire Wall	148~152
၂၄။	making, mounting, burning iso file and creating bootable usb from command line	153~155
၂၅။	Clean Memory and CPU Cache	156~157
၂၆။	For Linux System Administrator Interview	158~159
၂၇။	redhat linux ၊ subscription နှင့် ကြမယ်	160

linux မိတ်ဆက်

linux ဆိုတာ ဘာလဲ? သိသူလည်း ရှိမယ်။ မသိသူလည်း ရှိမယ်။ သိချင်နေသူတွေလည်း ရှိမယ်။ ဟုတ်ကဲ့။ ကျနော် သိသလောက် မျှဝေပါရအေ။

ခု office လုပ်ငန်းတွေ တော်တော်များများက computer တွေမှာ သုံးနေတဲ့ computer တွေမှာ install လုပ် ထည့်သွင်းထားတဲ့ operating system ကို Windows လို့ ခေါ်ပါတယ်။ windows xp, windows 7, windows 8, windows 10 စသည်ဖြင့်ပေါ့။ windows OS ရဲ့ ထုတ်လုပ်သူက microsoft ပါ။

အလားတူစွာပဲ၊ linux ဆိုတာသည်လည်း ဘက်စုံ သုံးလို့ရတဲ့ operating system platform တစ်ခုပါ။ ဒီ linux platform ပေါ်မှာမှ linux အသွင်ကဲ့တွေ အများကြီး ရှိပါတယ်။ ဒါတွေကို linux distro တွေလို့ ခေါ်ပါတယ်။

linux ကို ဘယ်နေရာမှာ မြင်နိုင်မလဲဆိုတော့၊ အကောင်းဆုံး ဥပမာပေးရရင်၊ ကျနော်တို့ mobile phone တွေမှာ အသုံးပြုထားတဲ့ Android ဆိုတာသည်လည်း linux OS တစ်ခုပါပဲ။ linux ရဲ့ဦးဆုံး ဖန်တီးရှင်ကတော့ Linus Torvalds ပါ။ 1991 ခုနှစ်မှာ စတင် ဖန်တီးခဲ့ပါတယ်။ linux ကို ပင်ဂျင်းနှင်းလေးနဲ့ ကိုယ်စားပြုလေ့ရှိပါတယ်။

နောက်ပိုင်းမှာ linux ဟာ hacker တွေ ကြားမှာ ရေပန်းစားသွားတာကြောင့်၊ အရှိန်အဟုံ့နဲ့ develope ဖြစ်လာပြီး၊ နယ်ပယ်အသီးသီးရဲ့ လိုအပ်ချက်အရ ဖြည့်စွက် ပြင်ဆင် ရေးသားထားတဲ့ linux distro တွေ အမြောက်အများ ပေါ်ထွေကဲပါတယ်။

linux ဟာ နည်းပညာလောကမှာ အရမ်း ပေါက်သွားတယ်လို့ ဆိုရမှာပါ။ linux distro ထောင်နဲ့ ချို့ပြီး ရှိပါတယ်။ အဲဒီထဲကမှ လူသိများ ထင်ရှားတာတွေကတော့ Debian, redhat, centOS, fedora, ubuntu, suse, kali, Arch,.. စသည်ဖြင့်ပေါ့။

linux ကို အသုံးပြုရာမှာ မိမိ လိုအပ်ချက်နဲ့ ကိုက်ညီမယ့် distro ကို ရွှေးချယ် အသုံးပြုရှုပါပဲ။ linux ဟာ windows တို့ mac OS တို့လို့ အခကြေးငွေ နဲ့ ဝယ်ယူရတာ မဟုတ်ပဲ၊ free အနေနဲ့ လွှတ်လပ်စွာ ဖြန့်ချိတဲ့ OS တစ်ခုပါ။ linux OS တွေကို internet ကနေ အလွယ်တကူ download လုပ် ရယူနိုင်ပါတယ်။

linux OS နဲ့ အတူ သူကို ဖန်တီးထားတဲ့ source code တွေကိုပါ တစ်ပါတည်း ထည့်ပေးထားတာကြောင့်၊ နည်းပညာရှင်တွေ developer တွေ အတွက် အများကြီး အကျိုး ရှိစေပါတယ်။ linux ကို personal သုံး desktop version အနေနဲ့ရော လုပ်ငန်းသုံး server version အနေနဲ့ပါ ရရှိနိုင်ပါတယ်။

security အားကောင်းတာ၊ ပေါ့ပါးမြန်ဆန်တာတွေကြောင့်၊ linux ကို bank တွေ telecom တွေ ISP (Internet Service Provider) တွေမှာ server အဖြစ် သုံးစွဲကြပါတယ်။ server အတွက် ထင်ရှားတဲ့ distro တွေကတော့။

1.Red Hat Enterprise Linux

2.Ubuntu Server

3.CentOS

4.SUSE Enterprise Linux

5.Fedora server စသည်ဖြင့် ရှိပါတယ်။

linux ကို personal အနေနဲ့လည်း အသုံးပြနိုင်ပြီး၊ ရုံးလုပ်ငန်း အတွက် Open office, photo editing အတွက် GIMP, design ဆွဲဖို့အတွက် Open CAD စတဲ့ application တွေ ရှိပါတယ်။ နောက်ပြီး linux ပေါ်မှာ windows application တွေကို wine ကနေ တစ်ဆင့် ထည့်သွင်း အသုံးပြုတာ DOTA II game ကစားတာတွေကိုလည်း ပြုလုပ်နိုင်ပါသေးတယ်။
personal အတွက် လူသုံး များတဲ့ distro တွေကတော့၊

1.Ubuntu

2.Fedora

3.Linux Mint

4.openSUSE

6.Deepin စသည်ဖြင့် ရှိပါတယ်။

ဒုံးအပြင်၊ Hacking အတွက် အထူးပြုထားတဲ့ linux distro တွေလည်း ရှိပါသေးတယ်။
Hacking အတွက် လူသုံး များတဲ့ distro တွေကတော့၊

1.Debian

2.Kali

3.Arch

4.Parrot စသည်ဖြင့် ရှိပါတယ်။

နောက်ပြီး linux ကို သုံးစွဲရင် virus အန္တရာယ် အတွက် ကြောက်စရာ မလိုပါဘူး။ error များ နည်းပါးပြီး system crash ဖြစ်ခြင်းမှုလည်း ကင်းဝေးပါတယ်။

ဒုံးကြောင့် linux ဟာ ရွှေးချယ် အသုံးပြုသင့်တဲ့ operating system platform တစ်ခုပါ။ နိုင်ငံတကာမှာတော့ linux ကို ကျယ်ကျယ်ပြန်ပြန် အသုံးပြုနေကြပါပြီ။

ကျနော်တို့ မြန်မာ နိုင်ငံမှာလည်း linux ကို စတင် အသုံးပြုနေကြပါပြီ။ linux ကို စိတ်ဝင်စားနေကြပါပြီ။

ဒုံးကြောင့် မိမိ လိုအပ်ချက်နဲ့ ကိုက်ညီမယ့် linux distro တစ်ခုကို ရွှေးချယ် အသုံးပြုကြပါလို့ ဒီကနေ အဆိုပြုလိုက်ပါရစေး။

linux အကြောင်း အကျဉ်းချုပ်ကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုကြပါစေး။

Yin Thu

(9-5-2018)



Yin

centOS linux

နည်းပညာလမ်းကြောင်းအရာ linux နဲ့ ပတ်သက်ပြီး လေ့လာမယ်ဆိုရင်၊ စမ်းသပ်မယ်ဆိုရင် centOS ကို virtual box (ကွဲပြားသော operating system များကို install လုပ် စမ်းသပ်နိုင်သော Application) မှာ တင်ပြီး လေ့ကျင့်တာ အသင့်တော်ဆုံးပါ။ ကျနော် suggestion ပေါ့နော်။ centOS နဲ့ red hat, fedora သိပ်မက္ခာခြားလုပ်သူး။ ubuntu, kali, susue တို့နဲ့လည်း အခြေခံ command တွေ သဘောတရားတွေ အတူတူပါပဲ။

ပြောစရာအနေနဲ့ centOS နဲ့ Red Hat ကွာတာကတော့ centOS က community ကို အခြေပြုပြီး red hat က license နဲ့ ဝယ်သုံးရတာပါပဲ။

RHEL (Red Hat Enterprise Linux) မှာ လိုအပ်ချက်အရာ packages တွေကို သွင်းမယ် ဆိုတာနဲ့ registration တန်းလုပ်ခိုင်းပါလိမ့်မယ်။ ဒါကြောင့် centOS ကို အသုံးပြုရတာက ပိုလွှတ်လပ်ပါတယ်။

ခုဆိုရင် centOS version 7 ကို ရောက်နေပါပြီ။ အရင် version အဟောင်းမှာ မပါခဲ့တဲ့ systemctl တို့လို့ command တွေလည်း ပါရှိလာပါပြီ။

centOS Community ENTerprise Operating System ကို Active multimedia, Excellent, IT bank စတဲ့ software အရောင်းဆိုင်တွေမှာ အလွယ်တကူ ဝယ်ယူနိုင်ပါတယ်။ online ကနေ download ချမယ်ဆိုရင်တော့ <https://www.centos.org/> ကနေ လွတ်လွတ်လပ်လပ် free ရယူနိုင်ပါတယ်။

ဒါကြောင့် နည်းပညာပိုင်းအရာ linux ကို လေ့လာမယ်ဆို centOS ကို အသုံးပြုပါ လို့ ဒီနေရာ ကနေ အဆိုပြုပါရမော်။

Yin Thu

(9-5-2018)



installation centos linux on virtual box

အများစုံ က linux ကို မကူးပြောင်းခွင့် microsoft ရဲ့ windows operating system နဲ့
အသားကျေ နေလေ့ ရှိပါတယ်။

ဒီအခါမှာ၊ မိမိ ရင်းနှီး ကျမ်းဝင်ပြီးသား windows ကို မဖြတ်ပဲ ORACLE virtual box ပေါ်မှာ
linux ကို သွင်းယူပြီး စမ်းသပ်တာက ပိုမို အဆင်ပြောမှုပါ။

ဒါက လေ့လာခါစမှာ ပိုအဆင်ပြောပါတယ်။

virtual box ကို download လုပ် ရယူရမယ့် link ကတော့ ဒီမှာပါ။

<http://download.virtualbox.org>

centos linux ကို virtual box ပေါ်ဘယ်လို သွင်းယူရမယ် ဆိုတာကိုတော့ အောက်ပါ link
က youtube video မှာ [ကြည့်ရနိုင်ပါတယ်။](#)

<https://www.youtube.com/watch?v=Pcl417NR2xc>

အားလုံး အဆင်ပြောပါစေ။

Yin Thu

(10-5-2018)



user, application, kernel and shell

linux အကြောင်းကို လေ့လာမယ်ဆိုရင် user, application, kernel နဲ့ shell အကြောင်းကို ဦးစွာ ပြောမှ ပြည့်စုံမှပါ။

#user_and_application

ရှင်းပါတယ်။ user ဆိုတာ computer ကို စေခိုင်း အသုံးပြုသူပေါ့။ application ဆိုတာ user အတွက် အလုပ် လုပ်ရတဲ့ software ပေါ့။

#kernel

နည်းပညာလောကမှာ kernel ဆိုတဲ့ အသုံးအနှစ်းကို ကြားဘူးမယ်ထင်ပါတယ်။ အားတော့ kernel ဆိုတာ ဘာလဲ လေ့လာကြည့်ရအောင်။

kernel ဆိုတာ computer operating system တစ်ခုရဲ့ မရှိမဖြစ် အချက်အချာအကျခုံး ပင်မ core program ပေါ့ ခေါ်ဆိုတဲ့ အမည်ပါ။

အဲဒီ core က ဘာတွေ လုပ်ပေးသလဲ ဆိုတော့ operating system ရဲ့ အစိတ်အပိုင်း အားလုံးအတွက် အခြေခံကျတဲ့ basic services တွေကို ထုတ်လုပ်ပေးပါတယ်။

application software တွေက ပေးပို့လိုက်တဲ့ input/output request တွေကို cpu (central processing unit) နဲ့ တစ်ခြား electronic အစိတ်အပိုင်းတွေ နားလည်တဲ့ machine language အဖြစ် ပြန်ဆို စီမံပေးပါတယ်။

ဒါကြောင့် computer operating system ရဲ့ အဓိက အကျခုံး ပင်မအပိုင်း core ကို kernel လို့ ခေါ်ပါတယ်။

#shell

kernel အကြောင်း ပြီးသွားတော့ shell အကြောင်း ပြောကြရအောင်။ shell ဆိုတာကတော့ computer အသုံးပြုသူ user နဲ့ kernel ရဲ့ အကြားက ကြားခံ interface တစ်ခုပါ။

operating system ထဲကို login ဝင်တယ်ဆိုပါစို့၊ user name နဲ့ password ကို စစ်ဆေးပြီးရင် login program ဟာ shell ကနေ တစ်ဆင့်မှ kernel ဆီ ချိုတ်ဆက်ရပါတယ်။

kernel ဟာ ပင်မ core ဖြစ်တာနဲ့ အညီ ငြင်းဆီကို instruction အမှားတွေ ရောက်သွားရင်、 operating system ပျက်စီးသွားနိုင်ပါတယ်။

shell ဟာ command line interpreter ဖြစ်ပြီး kernel ဆီကို instruction အမှားတွေ ရောက်မသွားအောင် ထိန်းကြောင်းပေးပါတယ်။

linux မှာ Alt+F2 နှင့် gnome-terminal လို့ ရိုက် Enter ခေါက်၊ terminal ကို ဖွင့်ပြီး မရှိတဲ့ command တစ်ခု ရိုက်ကြည့်မယ်ဆိုရင် ဥပမာ >> lulu လို့ ရိုက်ပြီး enter ခေါက်ကြည့်မယ်များ။ bash: lulu: command not found ဆိုပြီး ပေါ်လာပါလိမ့်မယ်။

ဒါက kernel ဆီ instruction အမှား ရောက်မသွားအောင် shell က တားဆီးလိုက်တာ ကြောင့်ပါပဲ။

shell နဲ့ ပတ်သက်လို့ linux system မှာ bash shell, sh shell, csh shell, tcsh shell စသဖြင့် ပါဝင်ပြီး bash shell ကို linux ရဲ့ default shell အနေနဲ့ ထည့်သွင်းထားလေ ရှိပါတယ်။ terminal မှာ ကျနော်တို့ မြင်ရမယ့် [root@linux /]# ဆိုတာဟာ bash shell ရဲ့ interface ပါပဲ။

terminal မှာ ကိုယ်အသုံးပြုနေတဲ့ shell အမျိုးအစားကို သိချင်ရင် echo \$0 လို့ ရှိကြပြီး ကြည့်နိုင်ပါတယ်။ ဒါကတော့ shell ရဲ့ အကြောင်းပါ။

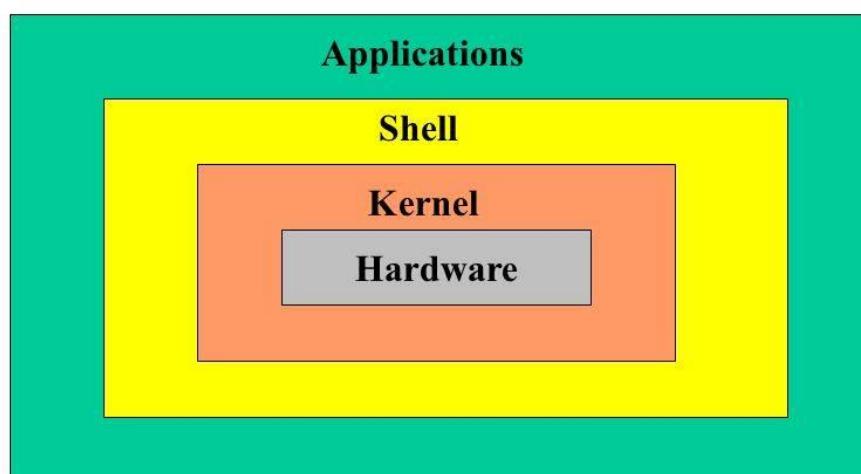
#summary

အားလုံးပေါင်းပြီး၊ အတိုချုပ်ပြောရရင်တော့ user ဆိုတာ computer ကို စေခိုင်းသူ၊ သူအတွက် kernel ဆီ instruction တွေ ပို့လွတ်ရတာသည် application ... kernel ဆိုတာက operating system တစ်ခုရဲ့ input/output တွေကို electronic အစိတ်အပိုင်း (hardware) တွေ နားလည်အောင် စီမံပေးတဲ့ ပင်မ core program ... shell ဆိုတာကတော့ user/application နဲ့ kernel ကြားက အမှားအမှန် စစ်ပေးတဲ့ filter တစ်ခုပေါ့။ user/application, kernel နဲ့ shell အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(10-5-2018)

User View of Linux Operating System



CentOS linux ကို laptop မှာ တင်ပြီးကြော်မယ် ထင်ပါတယ်။ CentOS linux os ကို root account နဲ့ log in ဝင်၊ Alt+F2 နှင့် gnome-terminal လို့ ရှိက် Enter ခေါက်လိုက်ရင် linux ခဲ့ command တွေကို run တဲ့ Terminal ပွင့်လာပါမယ်။

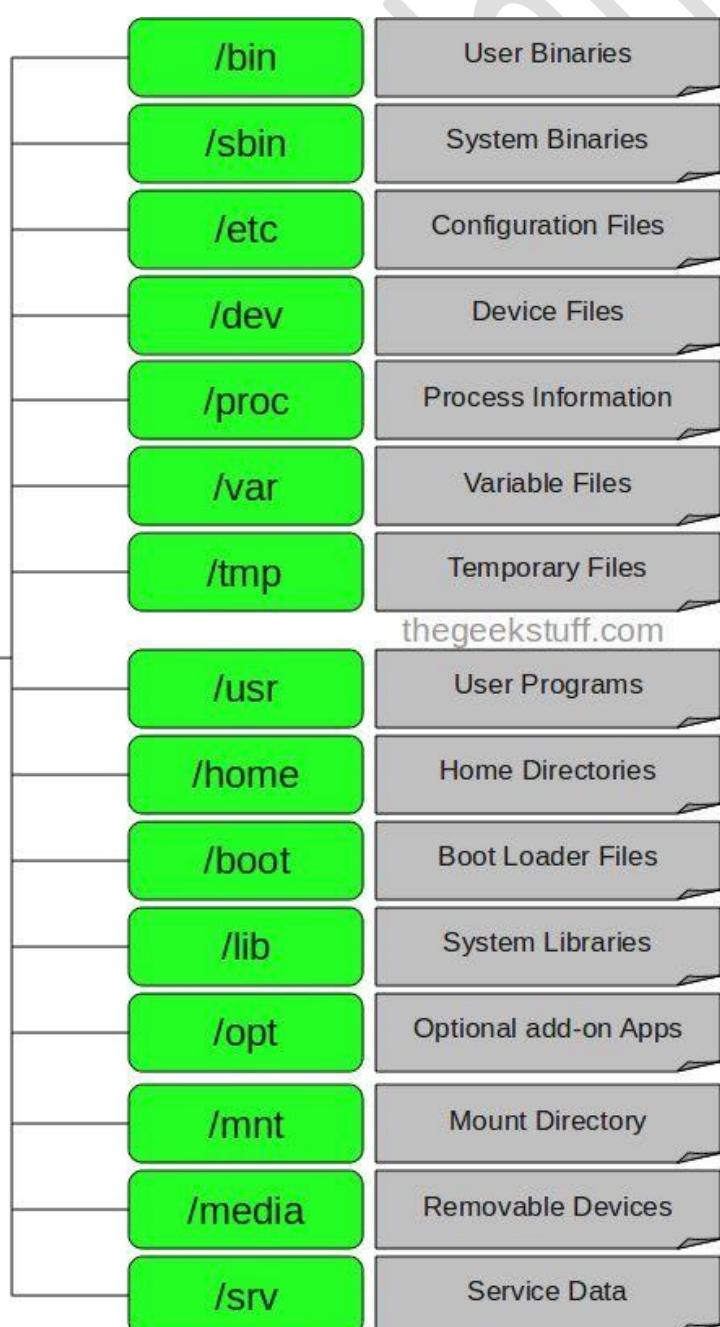
အဲဒီထဲမှာ cd / လို့ ရှိက် Enter ခေါက်လိုက်ရင် linux ခဲ့ပင်မ root directory ဆိုရောက်ပါမယ်။ ls လို့ ထပ်ရှိက် Enter ခေါက်လိုက်ရင် root, home, lib, etc, var, ... စသည်ဖြင့် directory တွေကို မြင်ရပါမယ်။

ခကာနေရင် အဲဒီ directory တွေဟာ ဘာအတွက်လဲ ဆိုတာကို ရှင်းပြထားတဲ့ Linux File System Hierarchy အကြောင်း တင်ပေးပါမယ်။

လေ့လာသူတွေ အဆင်ပြေပါစေ။

Yin Thu

(11-10-2018)



linux machine ၏ Terminal ဖွင့်ခြင်း

linux နဲ့ ရင်းနှီးသူတွေကတော့ ဒီခေါင်းစဉ်ကို ကြည့်ပြီး ရီကြမှာပါ။ ဟုတ်ကဲ ရီပါ။ ဒါပေမယ့် ခုမှ စလွှဲလာခါ သူ ၁ ယောက် အတွက်ကတော့ ဒါက လိုအပ်ပါတယ်။

windows operating system မှာ command တွေ ရိုက်ထည့်တဲ့ window ကို command prompt လို့ခေါ် ပြီး၊ linux မှာတော့ ဒါကို terminal လို့ ခေါ်ပါတယ်။

ဟုတ်ပြီဗျာ၊ user name, password ထည့်ပြီး linux machine ထဲ login စင်ပြီး၊ linux machine ချု desktop screen ပေါ်လာပြီ ဆိုကြပါစို့။

command တွေ ရိုက်ထည့်ဖို့ terminal လေးကို ဘယ်လို ဖွင့်မလဲ?

၁။ desktop ပေါ်မှာ mouse နဲ့ right click ထောက်လိုက်ပါ။ ပေါ်လာတဲ့ pop up menu ထဲကနေ Terminal သို့မဟုတ် open terminal ဆိုတာလေးကို ထပ်နှိပ်ပြီး terminal ကို ဖွင့်ပါ။

၂။ ဘယ်ဘက်အပေါ်ထောင့် application menu ထဲကနေ System Tools > Terminal ကို ရွေးပြီး ဖွင့်ပါ။

၃။ keyboard ပေါ်က Alter+F2 ကို တွဲ နှိပ်ပါ။ ပေါ်လာတဲ့ dialogue box ထဲမှာ gnome-terminal လို့ ရိုက် enter ခေါက်ပါ။

ဒီနည်းတွေထဲက တစ်နည်းနည်း အသုံးပြုပြီး Terminal ကို ဖွင့်နိုင်ပါပြီ။ ကဲ !! linux ကို စတင်လေ့လာမယ့်သူများ အဆင်ပြေကြပါစေ။

Yin Thu

(11-5-2018)



linux file system hierarchy

linux ရဲ command တွေ ရိုက်ထည့်ပေးရတဲ့ နေရာ၊ run တဲ့နေရာကို terminal လို့ ခေါ်ပါတယ်။ အဲဒီ terminal ထဲမှာ command တွေ မ run ခင် linux ရဲ file system တည်ဆောက်ပုံကို အရင်ဆုံး သိထားဖို့ လိုပါတယ်။

Terminal ထဲမှာ **cd /** လို့ ရိုက် Enter ခေါက်၊ **ls** လို့ ထပ်ရိုက် Enter ခေါက်လိုက်မယ်ဆိုရင် **bin, boot, dev, etc, home,** စသည်ဖြင့် directory/folder တွေကို မြင်ရပါလိမ့်မယ်။

ကိုယ့်ရဲ linux machine ထဲမှာ မြင်နေရတဲ့ အဲဒီ directory တွေဟာ ဘာအတွက်လဲဆိုတာ သိထားသင့်တာပေါ့နော်။ ပုံလေးနဲ့ တွဲပြီး လေ့လာကြည့်ရအောင်။

ပုံမှာ မြင်ရတဲ့အတိုင်း၊ linux ရဲ file system ကို tree ပုံစံနဲ့ တည်ဆောက်ထားပြီး ဒါကို file system hierarchy လို့ ခေါ်ပါတယ်။

ကဲ ! ပုံလေးနဲ့ တွဲပြီး စရှင်းလိုက်ရအောင်။

/ >> root directory လို့ ခေါ်ပါတယ်။ directory အားလုံးရဲ့ အထက်မှာ တည်ရှိပြီး၊ ကျွန်ုင်း အားလုံးဟာ root ရဲ အောက်မှာ တည်ရှိပါတယ်။ directory အားလုံးရဲ့ အချုပ်ပေါ့။

bin >> linux ကို run ဖို့ မရှိမဖြစ် လိုအပ်တဲ့ binary file တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

boot >> linux ကို boot လုပ် ဆွဲတင်ဖို့ လိုအပ်တဲ့ boot file တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

dev >> တပ်ဆင်ထားတဲ့ device တွေနဲ့ သက်ဆိုင်တဲ့ file တွေကို သိမ်းထားတဲ့ directory တစ်ခုပါ။

etc >> linux ပေါ်မှာ တင်ထားတဲ့ software တွေရဲ့ configuration file တွေကို သိမ်းထားတဲ့ directory တစ်ခုပါ။

home >> ဆောက်ထားတဲ့ user account တွေရဲ့ home directory ပါ။ ဥပမာ၊ ပုံထဲမှာ မြင်ရတဲ့ အတိုင်း၊ alice ဆိုတဲ့ user account ရှိရင် သူ့အတွက် home directory အောက်မှာ alice ဆိုတဲ့ directory လေး ဆောက်ပြီးသား ရှိနေမှာပါ။

root >> linux ရဲ super user ဖြစ်တဲ့ root account ရဲ home directory ပါ။ root account ဟာ သူ့ရဲ့ home directory ကို တွေ့ရှိနေတဲ့ account တွေလို့ home အောက်မှာ မဖန်တီးပါဘူး။ ပုံထဲကအတိုင်း / အောက်မှာ **root** ဆိုတဲ့ directory အဖြစ် သီးခြား ရပ်တည်ပါတယ်။

run >> boot လုပ်လိုက်တဲ့ အချိန်ကစလို့ run နေတဲ့ process တွေ၊ application တွေ ရဲ run time data တွေ ပါဝင်ပါတယ်။ reboot လုပ်လိုက်တဲ့ အခါမှာတော့ အဲဒီ data တွေကို ဖျက်ပစ်ပြီး အသစ်တစ်ဖန် ပြည်လည် စတင်ပါတယ်။

sbin >> system administration နဲ့ဆိုင်တဲ့ binary file တွေပါရှိတဲ့ directory တစ်ခုပါ။

tmp >> application တွက် run တဲ့အခါ ယာယီ run မယ့် file တခို့ကို သိမ်းထားတဲ့ directory တစ်ခုပါ။ အဲဒီ file တွေဟာ 10 ရက်အတွင်း ထပ်မံ access လုပ် မခံရရင် အလိုအလျောက် ဖျက်ပစ်ခံရမှာ ဖြစ်ပါတယ်။

usr >> user သွင်းယူထားတဲ့ software တွေ၊ user အသုံးပြုမယ့် command တွေ၊ system administration command တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

usr/bin (usr အောက်က bin) >> user နဲ့ သက်ဆိုင်တဲ့ command တွက် သိမ်းဆည်းထားတဲ့ directory တစ်ခုပါ။

usr/sbin (usr အောက်က sbin) >> user နဲ့ သက်ဆိုင်တဲ့ system administration command တွက် သိမ်းဆည်းထားတဲ့ directory တစ်ခုပါ။

usr/include (usr အောက်က include) >> program တွက် ထပ်မံ develope လုပ်ရာမှာ compile လုပ်ရန် လိုအပ်တဲ့ header file တွက် သိမ်းဆည်းထားတဲ့ directory တစ်ခုပါ။

usr/local (usr အောက်က local) >> လက်ရှိ user နဲ့သာ သီးခြားသက်ဆိုင်သော file တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

usr/lib (usr အောက်က lib) >> user တွေရဲ့ application တွေ အတွက် လိုအပ်တဲ့ library file တွက် သိမ်းဆည်းထားတဲ့ directory တစ်ခုပါ။

usr/share (usr အောက်က share) >> user တွေ share လုပ် ဝေမျှနှင့် အတွက် လိုအပ်သော file တွေ ရှိတဲ့ directory တစ်ခုပါ။

usr/src (usr အောက်က src) >> linux kernel ရဲ့ source code တွေ ပါရှိတဲ့ directory တစ်ခုပါ။

usr/X11R6 (usr အောက်က X11R6) >> X window system အတွက် လိုအပ်တဲ့ file တွေ ပါရှိတဲ့ directory တစ်ခုပါ။

var >> ပါဝင်တဲ့ အကြောင်းအရာတွေ မကြာခဏ dynamically ပြောင်းလဲနေတဲ့ variable data တွေ ရှိတဲ့ directory တစ်ခုပါ။ database, cache directories, log files စားတွေ ပါဝင်ပါတယ်။

var/log (var အောက်က log) >> system နဲ့ သက်ဆိုင်သော log file တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

var/spool (var အောက်က spool) >> စာပိုစနစ်၊ message ပိုစနစ်၊ printing စတာတွေနဲ့ တစ်ခြား queue လုပ်ရတဲ့ အလုပ်မှုနှင့်သမျပ်ဝါဝင်တဲ့ directory တစ်ခုပါ။

lib >> linux အတွက် လိုအပ်တဲ့ library file တွေ ပါဝင်ပါတယ်။ windows ရဲ့ dll file တွေလိုမျိုးပေါ့။

lost+found >> disk check လုပ်တဲ့အခါ တွေ့ရတဲ့ ပျက်စီးနေတဲ့ file တွေ၊ မည်သည့် directory နဲ့မ အဆက်အစပ် မရှိတဲ့ file တွေကို သိမ်းထားတဲ့ directory တစ်ခုပါ။ computer ကို စနစ်တကျ shutdown မလုပ်တဲ့အခါ အဲဒီလို file တွေ ရှိနေတတ်ပါတယ်။

misc >> linux နဲ့ သက်ဆိုင်တဲ့ အထွေထွေ file တွေ ပါရှိတဲ့ directory တစ်ခုပါ။

mnt >> CD Rom, Flash driv, printer နဲ့ တစ်ခြား ဆက်စပ် device တွေကို mount လုပ်တပ်ဆင်တဲ့ နေရာပါ။ media >> mount လုပ်ထားတဲ့ CD Drive ထဲက data တွေကို ဒီနေရာမှာ အလိုအလျောက် လာရောက် ဖော်ပြလေ့ ရှိပါတယ်။

opt >>ထပ်မံသွင်းယူထားတဲ့ optional software တွေ ပါဝင်တဲ့ directory တစ်ခုပါ။

proc >> pseudo-file system တွေ ပါရှိပါတယ်။ ဒီ file တွေဟာ system configuration parameters တွေပါ။ hard disk ပေါ်မှာ အမှန်တကယ် မတည်ရှိပဲ linux OS kernel ရဲ့ memory ပေါ်မှာပဲ ရှိနေတဲ့ file တွေ ဖြစ်ပါတယ်။ cpu information တွေ memory information တွေ ပါဝင်ပါတယ်။

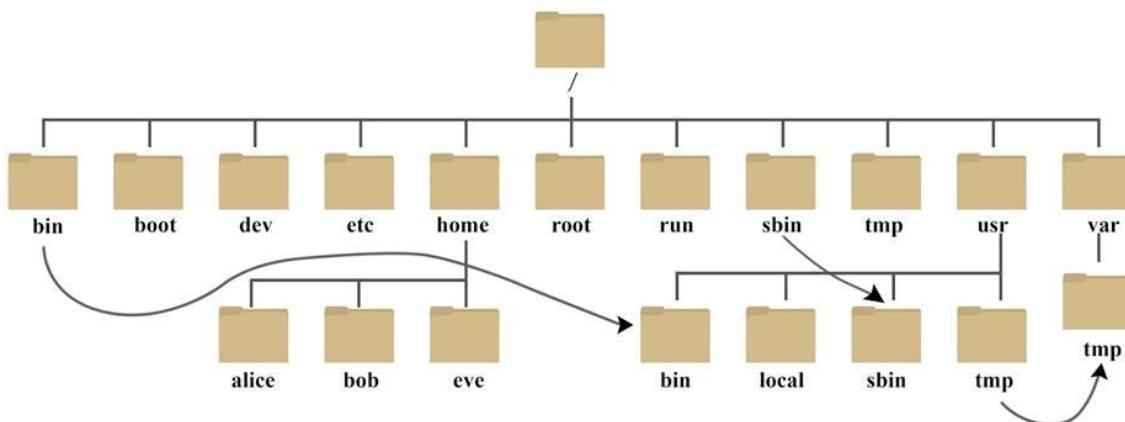
ခုလောက်ဆို linux ရဲ့ file system နဲ့ directory တစ်ခုချင်းစီရဲ့ အကြောင်းကို သိလောက်ပြီ ထင်ပါတယ်။ ဒီ directory တွေကို ကြည့်ချင်ရင် Terminalထဲမှာ ...

cd / လို့ ရိုက် enter ခေါက်၊ **ls** လို့ ထပ်ရိုက် enter ခေါက်၊ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(11-5-2018)



linux ရဲ command syntax အကြောင်း

linux machine ထဲကို root account နဲ့ log in ဝင်ပြီး terminal ကို ဖွင့်လိုက်တဲ့အခါ terminal မှာ ဥပမာအားဖြင့် [root@localhost~]# လို့မြင်ရပါလိမ့်မယ်။ အဲဒါကို shell prompt လို့ ခေါ်ပါတယ်။

၁။ **root** ကတော့ user name ပါ။

၂။ @ ကတော့ at ကို ညွှန်းဆိုတဲ့ သက်တပေါ့။

၃။ **localhost** ကတော့ linux ကို စတင် install လုပ်စဉ်မှာ ပေးခဲ့တဲ့ hostname (တစ်နည်းအားဖြင့် computer name) ပါ။ hostname ကို သိချင်ရင် **hostname** ဆိုတဲ့ command ကို သုံးပြီး ကြည့်နိုင်ပါတယ်။

၄။ ~ ကတော့ current account ရဲ့ home directory (/root) ကို ရောက်နေတယ် လို့ပြတာပါ။

၅။ # ကတော့ အခုံ terminal ဟာ root account က ဖွင့်ထားတာ လို့ သဘောရပါတယ်။ တစ္းဗုံး user account တွေမှာ ဆိုရင်တွေ \$ သက်တနဲ့ဖော်ပြပါတယ်။

ဒါကတော့ terminal ဖွင့်ဖွင့်ချင်းမြင်ရမယ့် အရာတွေပါ။ ဟုတ်ပြီ command အကြောင်း ဆက်သွားကြရအောင်။

command တစ်ခုမှာ

၁။ command

၂။ option

၃။ argument

ဆိုပြီး ၃ ပိုင်း ရှိပါတယ်။

ကဲ ! command တစ်ခုကို အစမ်း ရှိက်ကြည့်ကြပါစို့။

ls လို့ရှိက်ပြီး enter ခေါက်လိုက်ပါ။ directory ထဲမှာ ရှိတဲ့ အရာ တွေကို list ထုတ်ပြသပါလိမ့်မယ်။ hidden file တွေ directory တွေတော့ မပါဘူးပေါ့။

ဟုတ်ပြီ၊ hidden ဖြစ်နေတဲ့ အရာတွေကိုပါ ကြည့်ချင်တယ်။

ls -a -l လို့ရှိက်လိုက်ပါ။ hidden ဖြစ်နေတဲ့ အရာတွေကိုပါ မြင်ရပါလိမ့်မယ်။

ဒီနေရာမှာ ပြောချင်တာက **ls** သည် command ဖြစ်ပြီး **-a -l** သည် option ဖြစ်တယ် ဆိုတာပါပဲ။ option ဆိုတာ ကားဂိုယာလိုပါပဲ။ လိုအပ်ချက်အရ ဘယ်လမ်းကို ဘယ်ဂိုယာနဲ့ မောင်းမလဲ လိုအပ်တဲ့ ဂိုယာကို ထည့်ပေးရသလိုပေါ့များ။ မလိုအပ်ရင် option ကို ထည့်စရာ မလိုပါဘူး။

ကဲ ! ဟုတ်ပြီ။ argument အကြောင်း ဆက်သွားကြပါစို့။ အရင်ဆုံး **testfolder** ဆိုတဲ့ **folder** လေး ဆောက်ဖို့ **mkdir testfolder** လို့ရှိက် **Enter** ခေါက်လိုက်ပါ။

ls နဲ့ပြန်ကြည့်လိုက်ရင် **testfolder** ဆိုတဲ့ **folder/directory** လေး တိုးလာတာ မြင်ရပါမယ်။

အခါ folder လေးရဲ့ အချက်အလက် properties တွေကို **ls** နဲ့ ကြည့်ကြည့်ရအောင်။ ဒီလို ကြည့်ဖို့အတွက် **ls -l testfolder** လို့ ရှိက် **Enter** ခေါက်လိုက်ပါ။

ls -l testfolder ပေါ့။

ဒီနေရာမှာ -->

ls သည် **command**

-l သည် **option**

testfolder သည် **argument** ပါ။

ခုလောက်ဆို linux ရဲ့ command syntax ကို နားလည်ပြီ ထင်ပါတယ်။

application တစ်ခု သို့မဟုတ် command တစ်ခုဟာ ဘာလဲ သိချင်ရင် **whatis** နဲ့ ကြည့်နိုင်ပါတယ်။

ဥပမာ **whatis firefox**

ဒါ့အပြင် command တစ်ခုနဲ့ တစ်ခုကို semicolon ; ခြားပြီး တစ်ကြောင်းတည်း အနေနဲ့ run နိုင်ပါတယ်။ ဒီလို run ရင် ရလဒ် ခုကို အထက်အောက်ဖော်ပြပေးပါလိမ့်မယ်။

ဥပမာ **echo hello ; echo world**

command နှစ်ခုကို **&&** ဆိုတဲ့ operator ခြားပြီးတော့ တစ်ဆက်တည်း run နိုင်ပါတယ်။ ဒီလို run ရင် ပထမ command က အောင်မြင်မှု ဒုတိယ command က run ပါလိမ့်မယ်။

ဥပမာ **echo first && echo second** လို့ ရှိက် **Enter** ခေါက်ပါ။ ဒါဆို first second ဆိုတဲ့ result ရပါလိမ့်မယ်။ တယန် **zecho first && echo second** လို့ ရှိက် **Enter** ခေါက်ကြည့်ပါ။

bash: zecho: command not found ဆိုတဲ့ result ရပါလိမ့်မယ်။ အကြောင်းက **zecho** ဆိုတဲ့ command မရှိတဲ့ အတွက် မအောင်မြင်လို့ ဒုတိယ command ဖြစ်တဲ့ **echo second** သည်လည်း မ run တော့ပဲ **error result** ရလာတာပါ။

command နှစ်ခုကို **pipe** | သက်တဲ့ ခြားပြီးတော့ တစ်ဆက်တည်း run နိုင်ပါတယ်။ ဒီလို run ရင် ပထမ command က ရလာတဲ့ result ကို ဒုတိယ command က လက်ဆင့်ကမ်းယူပြီး run ပါလိမ့်မယ်။

ဥပမာ

၁။ မိမိ စက်မှာ install လုပ်ထားတဲ့ package တွေကို **yum list installed** ဆိုတဲ့ command နဲ့ စာရင်းထုတ်မယ်။

၂။ ရလာတဲ့ စာရင်း result ကို **more** ဆိုတဲ့ command နဲ့ ကြည့်မယ်။

ဒီလို အခြေအနေမျိုးမှာ **pipe** သက်တဲ့ | ခြားပြီး **yum list installed | more** လို့ တစ်ကြောင်းတည်း run နိုင်ပါတယ်။

yum list installed / more

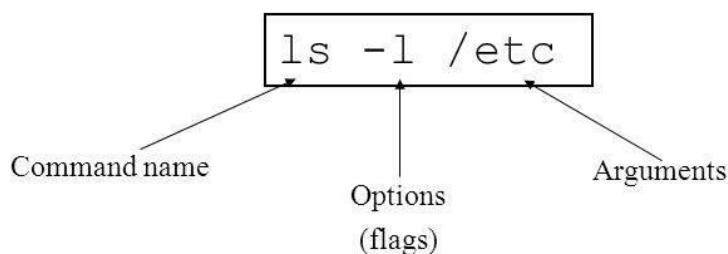
ကဲ ! ခုလောက်ဆို linux command တစ်ခုရဲ့ အခြေခံ တည်ဆောက်ပုံ နဲ့ တွေး သဘောတရား တရှိကို နားလည်နိုင်မယ်လို့ ယူဆပါတယ်။
အေးလုံး အဆင်ပြေပါစေ။

Yin Thu

(13-5-2018)

Linux Command Basics

⌘ To execute a command, type its name and arguments at the command line



file paths on linux system

ဒီတစ်ခါ linux နဲ့ ပတ်သက်လို့ အရှုံးရှင်းဆုံး topic လေးကို စလိုက်ရအောင်။ file path ဆိုတာ file တွေဆီသွားတဲ့ လမ်းကြောင်း အမျိုးအစားတွေကို ဆိုလိုတာပါ။ အရေးမကြီးပေမယ့် သိထားသင့်တဲ့ အချက်လေးပေါ့။

ဥပမာ >> **/root/Desktop/** (ဒါကို file path လို့ ခေါ်ပါတယ်)

file path အမျိုးအစား ၃ မျိုး ရှိပါတယ်။

၁။ absolute file path

root directory ရဲ့ သက်တာ / နဲ့ စတဲ့ file path ကို ပြောတာပါ။

cd /root/Desktop

၂။ relative file path

root directory ရဲ့ သက်တာ / နဲ့ မစတဲ့ ရှိုးရှိုး file path ကို ပြောတာပါ။

cd Desktop

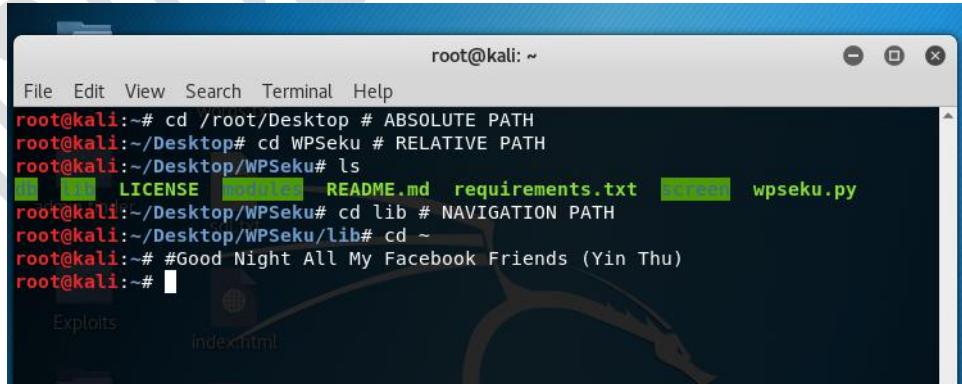
၃။ navigation file path

file ရှိရာဆိုကို command တစ်ကြောင်းတည်းနဲ့ မသွားပဲ directory တွေ တစ်ဆင့်ပြီးမှတစ်ဆင့် ရှိက်ပြီး သွားတာကို ဆိုလိုတာပါ။

ဥပမာ၊ **root** အောက်က **Desktop** အောက်က **WPSeku** အောက်က **lib** ဆိုတဲ့ **folder** ဆိုကို သွားမယ် ဆိုပါတယ့်။

cd /root/Desktop/WPSeku/lib လို့ တစ်ကြောင်းတည်း မရှိက်ပဲ အောက်မှာ ဖော်ပြထားသလို တစ်ဆင့်ပြီးမှတစ်ဆင့် သွားတာကို ဆိုလိုတာပါ။

cd /root
cd Desktop
cd WPSeku
cd lib



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# cd /root/Desktop # ABSOLUTE PATH
root@kali:~/Desktop# cd WPSeku # RELATIVE PATH
root@kali:~/Desktop/WPSeku# ls
LICENSE modules README.md requirements.txt wpseku.py
root@kali:~/Desktop/WPSeku# cd lib # NAVIGATION PATH
root@kali:~/Desktop/WPSeku/lib# cd ~
root@kali:~# #Good Night All My Facebook Friends (Yin Thu)
root@kali:~#
```

ကဲ ! file path တွေ အကြောင်းကတော့ ဒီလောက်ပါပဲ။ ရှင်းရှင်းလင်းလင်းပါပဲနော်။
အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(13-5-2018)

linux စက်ပေါ်တွင် directory တစ်ခုမှတစ်ခုသို့ ကူးပြောင်းခြင်း

ဒီတခါး linux ရဲ့ terminal ပေါ်ကနေ **directory/folder** တစ်ခုကနေ အောက်တစ်ခုဆို ကူးပြောင်း ကြည့်ရှုတဲ့ အပိုင်းကို လေ့လာကြဖို့။

၁။ linux မှာ **directory** တစ်ခုကနေ တစ်ခု change ဖို့ ကူးပြောင်းဖို့ အတွက် terminal ထဲကနေဆိုရင် **cd** ဆိုတဲ့ command ကို သုံးပြီး ကူးပြောင်းရပါတယ်။ **cd** ရဲ့ အဓိပါယ်က **change directory** ပါ။ **cd**

ဥပမာ **home directory** ထဲကနေ **root** အောက်က **var** အောက်က **log** ဆိုတဲ့ **directory** ဆို သွားချင်တယ်ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါပဲ။ **cd /var/log**

၂။ လက်ရှိ ရောက်ရှိရာ နေရာကနေ **user** ရဲ့ မူလ **home directory** ကို တန်းသွားချင်ရင် **cd** ကို သုံးပါတယ်။ **cd ~** လိုလည်း သုံးနိုင်ပါတယ်။ **cd** or **cd ~**

၃။ **root directory** ကို သွားချင်ရင် **cd /** ကို သုံးပါတယ်။ **cd /**

၄။ **current directory** ကို ဖော်ပြုချင်ရင် . သို့မဟုတ် ./ ကို သုံးတယ်။ ~+ လိုလည်း သုံးနိုင်ပါတယ်။ **cd .** or **cd ./** or **cd ~+**

၅။ လက်ရှိ ရောက်ရှိရာ နေရာကနေ **parent directory** ကို တစ်ဆင့်ပြန်တက်ချင်ရင် .. ကိုသုံးပါ တယ်။ **cd ..**

၆။ အထက်မှာ ရှိတဲ့ **dir** ၂ ဆင့်ကို ပြန်တက်ချင်ရင် .../.. ကို သုံးပါတယ်။ **cd ../../**

၇။ **previous working directory** ကို သွားချင်ရင် **cd -** ကို သုံးတယ်။ လက်ရှိ ရောက်ရှိနေတဲ့ **directory** မတိုင်ခင်က ဝင်ခဲ့တဲ့ **directory** ဆိုကို ပြန်ရောက်သွားမှာပါ။ **cd -** အစား: **cd ~-** လိုလည်း သုံးနိုင်ပါတယ်။ **cd -** or **cd ~-**

၈။ ကိုယ် ဘယ်ကို ရောက်နေလဲ? ကိုယ် ရောက်ရှိနေတဲ့ **directory** လမ်းကြောင်းကို ကြည့်ချင်ရင် **pwd** ဆိုတဲ့ command ကို သုံးပြီး သိနိုင်ပါတယ်။ **pwd**

လွယ်လွယ်လေးပါ မခက်ပါဘူးနော်။

ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(13-5-2018)

Linux cd Command

- The "cd" stands for 'change directory' and this command is used to change the current directory i.e; the directory in which the user is currently working.

Syntax:

- **cd <dirname>**

Example:

- **cd certs**

file နှင့် directory များကို list ထုတ် ကြည့်ခြင်း

linux machine ထဲကို root account နဲ့ log in ဝင်၏ terminal ကို ဖွင့်ပြီးပြီ ဆိုပါစိုး။ centos linux ရဲ့ terminal မှာ [root@localhost~]# လို့မြင်ရပါမယ်။

ဒီတစ်ခါ ရောက်ရှိနေတဲ့ **directory** အောက်မှာ ဘာတွေ ရှိလဲကြည့်ရအောင်။ ဒီလို **ကြည့်ဖို့အတွက် ls ဆိုတဲ့ command** ကို သုံးပါတယ်။ ls command က **current directory** အောက်မှာ ရှိတဲ့ **sub directory** နဲ့ **file** တွေကို ပြသပေးပါတယ်။

၁။ **ls** လို့ ရှိက်ပြီး **enter** ခေါက်တဲ့အခါ **directory** အောက်မှာ ရှိတဲ့ **file** တွေ **directory** တွေကို မြင်ရပါလိမ့်မယ်။ **ls**

၂။ ဒါပေမယ့် **hidden** ပေးထားတဲ့ **file** တွေ **directory** တွေကို မြင်ရမှာ မဟုတ်ပါဘူး။ **hidden** ဖြစ်နေတာတွေကို မြင်ရဖို့အတွက် -a ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ **ls -a** လို့ ရှိက် **enter** ခေါက်ကြည့်လိုက်ပါ။ **name** ရှေ့မှာ **dot** အစက်လေးတွေ ပါတဲ့ **hidden file** တွေကို မြင်ရပါမယ်။ **ls -a**

၃။ **ls** နဲ့ **ကြည့်တဲ့အခါ ကျလာတဲ့ list** ထဲမှာ (.) ဆိုတဲ့ **current directory** ကို ညွှန်းတဲ့ အစက်လေး ၁ ခု၊ (..) ဆိုတဲ့ **parent directory** ကို ညွှန်းတဲ့ အစက်လေး J ခု ကို မြင်ရပါမယ်။ ဒါတွေကို မမြင်ချင်ဘူးဆိုရင် **ls -A** ကို သုံးနိုင်ပါတယ်။ **ls -A**

၄။ **file** တွေ **directory** တွေကို ပေးထားတဲ့ **read, write, execute (rwx)** permission တွေ၊ **access, modify, change time** တွေနဲ့ တွဲလျက် မြင်ချင်ရင်တော့ -l ဆိုတဲ့ **option** ကို ထပ်ဖြည့်ပေးရပါမယ်။ **ls -l**

၅။ **file** တွေ ဆောက်ခဲ့တဲ့ အချိန်ပေါ်မှတည်ပြီး အစဉ်လိုက် ပြသစေချင်ရင်တော့ -t ဆိုတဲ့ **option** ကို သုံးပါတယ်။ **ls -l -t**

၆။ **directory** တစ်ခုကို ကြည့်တဲ့အခါ သူတို့ရဲ့ **sub directory** တွေကိုပါ တစ်ခါတည်း မြင်ချင်တယ်ဆိုရင် -R ဆိုတဲ့ **option** ကို သုံးပါတယ်။ **ls -R**

၇။ **file** တစ်ခုတည်းကို အထူးပြုပြီး ကြည့်ချင်တဲ့အခါ ဥပမာ၊ **ls -l filename** လို့ သုံးစွဲ ရပါတယ်။ **ls -l filename**

၈။ ဒါက လက်ရှိ **current directory** အောက်က **file** ကို ကြည့်တာပါ။ တစ်ခြားနေရာမှာ ရှိတဲ့ **file** ကို ကြည့်မယ်ဆိုရင်တော့ အဲဒီ **file** တည်ရှိရာ လမ်းကြောင်း **absolute path** ကို ထည့်ပေးရပါမယ်။ ဥပမာ၊ **root directory** အောက်က **etc** ဆိုတဲ့ **directory** အောက်က **passwd** ဆိုတဲ့ **file** ကို လုမ်းကြည့်မယ် ဆိုပါစိုး။ **ls -l /etc/passwd**

၉။ **directory** တစ်ခုတည်းကို အထူးပြုပြီး ကြည့်ချင်တဲ့အခါ ဥပမာ လက်ရှိ ရောက်ရှိနေတဲ့ **current directory** ထဲက **testfolder** ဆိုတဲ့ **directory** ကို အထူးပြုပြီး ကြည့်မယ်ဆိုပါစို့။ **ls -l -d testfolder** လို့ သုံးစွဲရပါတယ်။ **ls -l -d testfolder**

၁၀။ တစ်ခြားနေရာမှာ ရှိတဲ့ **directory** ကို ကြည့်မယ်ဆိုရင်တော့ အဲဒီ **directory** တည်ရှိရာ လမ်းကြောင်း **absolute path** ကို ထည့်ပေးရပါမယ်။ ဥပမာ ၁၂။ **root directory** အောက်က **var** ဆိုတဲ့ **directory** အောက်က **log** ဆိုတဲ့ **directory** ကို လုမ်းကြည့်မယ် ဆိုပါစို့။ **ls -l -d /var/log** ပေါ့။ **ls -l -d /var/log**

၁၁။ **file/folder** တစ်ခုရဲ့ **properties** တွေကို **ls** နဲ့ ကြည့်ရာမှာ **human readable format** နဲ့ ပြုသစေလိုတဲ့အခါ **ls -lh** ကို သုံးရပါတယ်။ **ls -lh testfolder**

၁၂။ **file/folder** တွေကို (,) **comma** ဗျားပြီး ပြုသစေချင်တဲ့အခါ **-m** ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ **ls -m**

၁၃။ **file/folder** တွေရဲ့ **inode number** တွေအရ ဘရင်းထုတ် ပြုသစေလိုတဲ့အခါ **-i -l** ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ **ls -i -l**

၁၄။ **file/folder** တွေကို **size** အရ ဖော်ပြုစေလိုတဲ့အခါ **-S** ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ **ls -ls**

၁၅။ **file/folder** တွေကို **visual classification** နဲ့ ပြုသစေလိုတဲ့အခါ **-F** ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ **ls -F**

option တွေကို **ls command** နဲ့ တစ်ခါတည်းတွဲစပ်ပြီးလည်း သုံးနိုင်ပါတယ်။ အောက်ပါ အတိုင်းပေါ့။ **ls -alt** သို့မဟုတ် **ls -a -l -t** သို့မဟုတ် **ls -alF** စသည်ဖြင့်ပေါ့။

ကဲ ! ဒီလောက်ဆို **ls** ဆိုတဲ့ **command** ကို အသုံးပြုပြီး၊ **file** တွေ **directory** တွေကို ဘယ်လို ကြည့်ရမလဲ သိပြုထင်ပါတယ်။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(14-5-2018)

File Type	# of Hard Links	User Permissions	Group Permissions	Other Permissions	Size	Date	File or Directory Name
drwxr-xr-x	4	devnet	devnet	4096	2009-09-28 05:13	Desktop	
drwxr-xr-x	6	devnet	devnet	4096	2009-09-25 07:23	Documents	
drwxr-xr-x	49	devnet	devnet	4096	2009-09-25 07:23	Music	
drwxr-xr-x	2	devnet	devnet	4096	2009-09-25 07:11	Network	
drwxr-xr-x	2	devnet	devnet	4096	2009-09-25 07:04	Pictures	
drwxr-xr-x	2	devnet	devnet	4096	2009-09-25 07:11	Public	
drwxr-xr-x	2	devnet	devnet	4096	2009-09-25 07:11	Templates	
drwxr-xr-x	2	devnet	devnet	4096	2009-09-25 07:11	Videos	

d - directory
r - readable
w - writeable
x - executable

file နှင့် သက်ဆိုင်သော

linux နဲ့ ပတ်သက်လို ဒီတစ်ခါတော့ file တွေနဲ့ သက်ဆိုင်တဲ့ အကြောင်းလေး ပြောကြရအောင်။ တို့တို့ရှင်းရှင်းပဲ စလိုက်ကြတာပေါ့။

၁။ **file** တစ်ခုဟာ ဘာအမျိုးအစားလဲ သိချင်ရင် **file** ဆိုတဲ့ **command** နဲ့ ကြည့်နိုင်ပါတယ်။ ဥပမာ **file1.txt** ဟာ ဘာအမျိုးအစားလည်း သိချင်တယ် ဆိုပါစို့။ **file file1.txt** လို့ရှိက် enter ခေါက်ရုံ ပါပဲ။ **file file1.txt**

၂။ **file** တစ်ခုကို တည်ဆောက်ချင်ရင် **touch** ဆိုတဲ့ **command** နဲ့ တည်ဆောက်နိုင်ပါတယ်။ ဥပမာ **file1.txt** ဆိုတဲ့ **text file** လေး တစ်ခု တည်ဆောက်ချင်တယ် ဆိုပါစို့။ **touch file1.txt** လို့ရှိက် enter ခေါက်ပေါ့။ **touch file1.txt**
ပြီးရင် **ls** ဆိုတဲ့ **command** လေးနဲ့ ပြန်ကြည့်လိုက်တဲ့ အခါ ကျင်ရှိတာလေးကို မြင်ရပြီပေါ့။ **ls**

၃။ အမည်တူပြီး **no.** နံပါတ်စဉ် မတူတဲ့၊ တစ်ခုထက် ပိုတဲ့ **file** တွေကို **command** ၁ ကြောင်း တည်းနဲ့ တည်ဆောက်မယ် ဆိုပါစို့။ **touch filename{1..5}** ဆိုတဲ့ **format** ကို အသုံးပြုရပါတယ်။ နံပါတ်စဉ်တွေကို တွေ့နိုင်း **brace** ထဲ ထည့်ထားလိုက်တဲ့ သဘောပါ။
1 ကနေ **5** အထိ **file** ငါးခု အစဉ်လိုက် ဆောက်မယ်ဆိုရင် **{1..5}**။ **1** ကနေ **10** အထိ **file 10** ခု အစဉ်လိုက် ဆောက်မယ်ဆိုရင် **{1..10}**။

ဥပမာ **file1** ကနေ **file10** အထိ နံပါတ်စဉ်ထိုးထားတဲ့ **.txt** နဲ့ ဆုံးတဲ့ **file 10** ခု ဆောက်မယ် ဆိုပါစို့။ **touch file{1..10}.txt** လို့ရှိက် enter ခေါက်ရုံပါပဲ။ **touch file{1..5}.txt**
ပြီးရင် ရလဒ်ကို **ls** နဲ့ ပြန်ကြည့်လိုက်ပါရှိး။ **ls**

၄။ **file** သို့မဟုတ် **directory** တစ်ခုရဲ့ **name** မှာ **space** ထည့်ပြီး ဆောက်ချင်ရင် **filename** ရဲ့ပထမပိုင်း အဆုံးမှာ **backslash ** ကို ထည့်ပြီး သုံးရပါတယ်။
ဥပမာ **file one** ဆိုတဲ့ အမည်နဲ့ **file** နဲ့ **one** ကြားမှာ **space** ခြားပြီး **text file** တစ်ခု တည်ဆောက် မယ်ဆိုပါစို့။ **touch file\ one.txt** လို့ရှိက် enter ခေါက်ရုံပေါ့။ **touch file\ one.txt**

၅။ ဒီတစ်ခါ **ls command** နဲ့ ခေါ်ကြည့်လို့ မမြင်နိုင်ပဲ **ls -a** လို့ ကြည့်မှ မြင်နိုင်မယ့် **hidden file** လေးတွေ ဆောက်ကြည့်ရအောင်။ လွယ်ပါတယ်နော်။

ဆောက်လိုက်တဲ့ **file name** ရဲ့ ရှေ့ဆုံးမှာ **dot (.)** အစက်လေး ၁ ခု ထည့်ပေးလိုက်ရင် **hidden file** ဖြစ်သွားပါမယ်။ ဥပမာ **file1.txt** ကို **hidden** အနေနဲ့ ဆောက်ချင်တယ် ဆိုပါစို့။ **touch .file1.txt** လို့ ရှိက် enter ခေါက်ရုံပေါ့။ **touch .file1.txt** ဆောက်ပြီးရင် **ls command** နဲ့ ခေါ်ကြည့်လိုက်ပါ။ **.file1.txt** ဟာ **list** ထဲမှာ ပါမလာတော့ပါဘူး။ **ls**

ls -a လို့ **-a** ဆိုတဲ့ **option** လေးခံပြီး ထပ် ခေါ်ကြည့်ပါမြို့။ **list** ထဲမှာ **.file1.txt** လေး ပြန်ပါလာ တာကို တွေ့ရပါလိမ့်မယ်။ **ls -a**

၆။ **file** တစ်ခု ရဲ့ အထဲမှာ ပါဝင်တဲ့ **size** ကို သိချင်ရင်တော့ **wc** ဆိုတဲ့ **command** နဲ့ ကြည့်ရပါတယ်။ **wc command** မှာ **-c -l -w** ဆိုပြီး **options** ၃ ခု ရှိတယ်။

-c က **character** အရေအတွက်၊ **-l** က **line** အရေအတွက်၊ **-w** က **word** အရေအတွက်။ ဥပမာ **file1.txt** ရဲ့ အထဲမှာ ပါဝင်တဲ့ **size** ကို သိချင်တယ် ဆိုပါစို့။ **wc file1.txt** လို့ ရှိက် enter ခေါက်ရုံပါပဲ။ **wc file1.txt**

ဒီထဲကမှ **character** တစ်ခုတည်း ကွက်ကြည့်ချင်ရင်တော့ **wc -c file1.txt** ပေါ့။

wc -c file1.txt

ဒီထဲကမှ **line** အရေအတွက် တစ်ခုတည်း ကွက်ကြည့်ချင်ရင်တော့ **wc -l file1.txt** ပေါ့။

wc -l file1.txt

ဒီထဲကမှ **word** အရေအတွက် တစ်ခုတည်း ကွက်ကြည့်ချင်ရင်တော့ **wc -w file1.txt** ပေါ့။

wc -w file1.txt

၇။ **file** တစ်ခု ရဲ့ **actual size** ကို သိချင်ရင်တော့ **ls -lh** ဆိုတဲ့ **command** နဲ့ ကြည့်ရပါတယ်။

ဥပမာ **file1.txt** ရဲ့ **size** ကို သိချင်တယ် ဆိုပါစို့။ **ls -lh** လို့ ရှိက် enter ခေါက်ရုံပါပဲ။ ကျလာတဲ့ **list** ထဲမှာ **file1.txt** ရဲ့ **size** ကို မြင်ရပါမယ်။ **ls -lh**

၈။ **file** ကို ဖျက်ချင်တဲ့ အခါ **rm** ဆိုတဲ့ **comand** ကို သုံးပြီး ဖျက်နိုင်ပါတယ်။

ဥပမာ **file1.txt** ဆိုတဲ့ **file** ကို ဖျက်မယ် ဆိုပါစို့။ **rm file1.txt** လို့ ရှိက် enter ခေါက်ရုံပေါ့။ ဖျက်မှာ သေချာလား **yes** or **no** မေးပါလိမ့်မယ်။ ဒီ အခါ **y** ကို ထပ်ရှိက်ပြီး **enter** ခေါက် လိုက်ရုံပါပဲ။ **rm file1.txt**

၉။ **file** တစ်ခုကို ဖျက်တဲ့ အခါ “ဖျက်မှာ သေချာလား” **yes** or **no** ဘာမှ မမေးပဲ တန်းပြီး ဖျက်စေ ချင်ရင်တော့၊ **rm** ကို **-rf** ဆိုတဲ့ **option** နဲ့ တွဲသုံးပေးရပါတယ်။ ဥပမာ **file1.txt** ကို ဖျက်မယ် ဆိုပါစို့။ **rm -rf file1.txt** ပေါ့။ **rm -rf file1.txt**

file တွေနဲ့ သက်ဆိုင်တဲ့ အကြောင်းတွေကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(14-5-2018)

```

root@kali:~# touch file1.txt #CREATING A FILE
root@kali:~# echo "Hello! world" >> file1.txt
root@kali:~# wc file1.txt
2 2 file1.txt
root@kali:~# ls -lh
total 36K
drwxr-xr-x 7 root root 4.0K May 13 09:40 Desktop
drwxr-xr-x 2 root root 4.0K Aug 2 2017 Documents
drwxr-xr-x 1 root root 4.0K Aug 3 2017 Downloads
-rw-r--r-- 1 root root 23 May 14 00:55 file1.txt
drwxr-xr-x 2 root root 4.0K Aug 2 2017 Music
drwxr-xr-x 2 root root 4.0K Aug 3 2017 Pictures
drwxr-xr-x 2 root root 4.0K Aug 3 2017 Public
drwxr-xr-x 2 root root 4.0K Aug 2 2017 Templates
drwxr-xr-x 2 root root 4.0K Aug 3 2017 Videos
root@kali:~# HAVE A NICE DAY ALL MY FACEBOOK FRIENDS (Yin Thu)
root@kali:~#

```

directory နှင့် သက်ဆိုင်သော

file တွေ သိမ်းထားတဲ့ အနေဖြင့် **folder/directory** လို့ ခေါ်ပါတယ်။ ကဲ ! **directory** နဲ့ သက်ဆိုင်တာ လေးတွေကို ပြောကြုံရအောင်။

၁။ **directory** တစ်ခု ဆောက်ချင်ရင် **mkdir** ဆိုတဲ့ **command** ကို သုံးပါတယ်။ ဥပမာ **testfolder** လို့ အမည်ရတဲ့ **directory** တစ်ခု ဆောက်ချင်တယ် ဆိုပါစို့။ **mkdir testfolder** လို့ ရှိက် enter ခေါက်ရုံပေါ့။ **mkdir testfolder**

ရလဒ်ကို **ls** နဲ့ ပြန်ကြည့်လိုက်ပါဉီး။ **current location** ရဲ့ အောက်မှာ **testfolder** ဆိုတဲ့ **directory** အသစ်ကို တွေ့ရပါလိမ့်မယ်။ **ls**

၂။ file တွေ တည်ဆောက်ရာမှာ သုံးသလိုပါပဲ၊ အမည်တူပြီး **no.** နံပါတ်စဉ် မတူတဲ့၊ တစ်ခု ထက်ပိုတဲ့၊ **directory** တွေကို **command** ၁ ကြောင်းတည်းနဲ့ တည်ဆောက်မယ် ဆိုပါစို့။ **mkdir directoryname{1..5}** ဆိုတဲ့ **format** ကို အသုံးပြုရပါတယ်။ နံပါတ်စဉ်တွေကို တွန့်ကွင်း **brace** ထဲ ထည့်ထားလိုက်တဲ့ သဘောပါ။

၁ ကနေ 5 အထိ **directory** ငါးခု အစဉ်လိုက် ဆောက်မယ်ဆိုရင် **{1..5}**။

၁ ကနေ 10 အထိ **directory** ဆယ်ခု အစဉ်လိုက် ဆောက်မယ်ဆိုရင် **{1..10}**။

ဥပမာ **folder** အမည်နဲ့ ၁ ကနေ 5 အထိ၊ နံပါတ်စဉ်ထိုးထားတဲ့ **directory** ၅ ခု ခဲ့ ဆောက်မယ် ဆိုပါစို့။ **mkdir folder{1..5}** လို့ ရှိက် enter ခေါက်ရုံပေါ့။ **mkdir folder{1..5}**

ပြီးရင် ရလဒ်ကို **ls** နဲ့ ပြန်ကြည့်လိုက်ပါဉီး။ **ls**

၃။ အမည် မတူတဲ့ **directory** အများကြီးကို **command** တစ်ကြောင်းတည်းနဲ့ ဆောက်နိုင်ပါ သေးတယ်။ အောက်ပါအတိုင်းပေါ့။ **mkdir tom jerry mickey**

ls နဲ့ ပြန်ခေါ်ကြည့်လိုက်ရင် **current location** အောက်မှာ **tom, jerry, mickey** ဆိုတဲ့ **directory** သုံးခု တိုးလာတာ တွေ့ရပါမယ်။

၄။ **directory** တစ်ခု ဆောက်ပြီး အဲဒီ **directory** အောက်မှာ **subdirectory** တွေကိုပါ တဝါတည်း **command** တစ်ကြောင်းတည်းနဲ့ ထည့်ဆောက်သွားချင်ရင် **mkdir** ကို **-p** ဆိုတဲ့ **option** နဲ့ တွဲသုံးပေးရပါတယ်။ ဥပမာ **testfolder** ဆိုတဲ့ **directory** နဲ့ အတူ သူ့အောက်မှာ **sub1** ဆိုတဲ့ **directory**၊ **sub1** အောက်မှာ **sub2** ဆိုတဲ့ **directory** ကိုပါ **command** တစ်ကြောင်း တည်းနဲ့ ဆောက်မယ် ဆိုပါစို့။ **mkdir -p testfolder/sub1/sub2** လို့ ရှိက် enter ခေါက်ရုံပေါ့။

mkdir -p testfolder/sub1/sub2

ပြီးရင် မိမိ ဆောက်လိုက်တာ အောင်မြင်လား၊ မအောင်မြင်လားကို **tree** ဆိုတဲ့ **command** သုံးပြီး ပြန်စစ်နိုင်ပါတယ်။ **tree testfolder**

၅။ **directory** ဆောက်ရင်း **permission** ပါ သတ်မှတ်ပေးဖို့ လိုအပ်ရင် **-m** ဆိုတဲ့ **option** ကို သုံးရပါတယ်။ ဥပမာ **testfolder** ဆိုတဲ့ **directory** ကို ဆောက်ရင်း **permission (read, write, execute)** သတ်မှတ်ပေးချင်တယ် ဆိုပါစို့။ **mkdir -m 750 testfolder** လို့ ရိုက် **enter** ခေါက်ရုံပါပဲ။ **mkdir -m 750 testfolder**

(**Permission** အကြောင်းကို နောက်လာမယ့် သင်ခန်းစာမှာ ဖေါ်ပြပါမယ်)

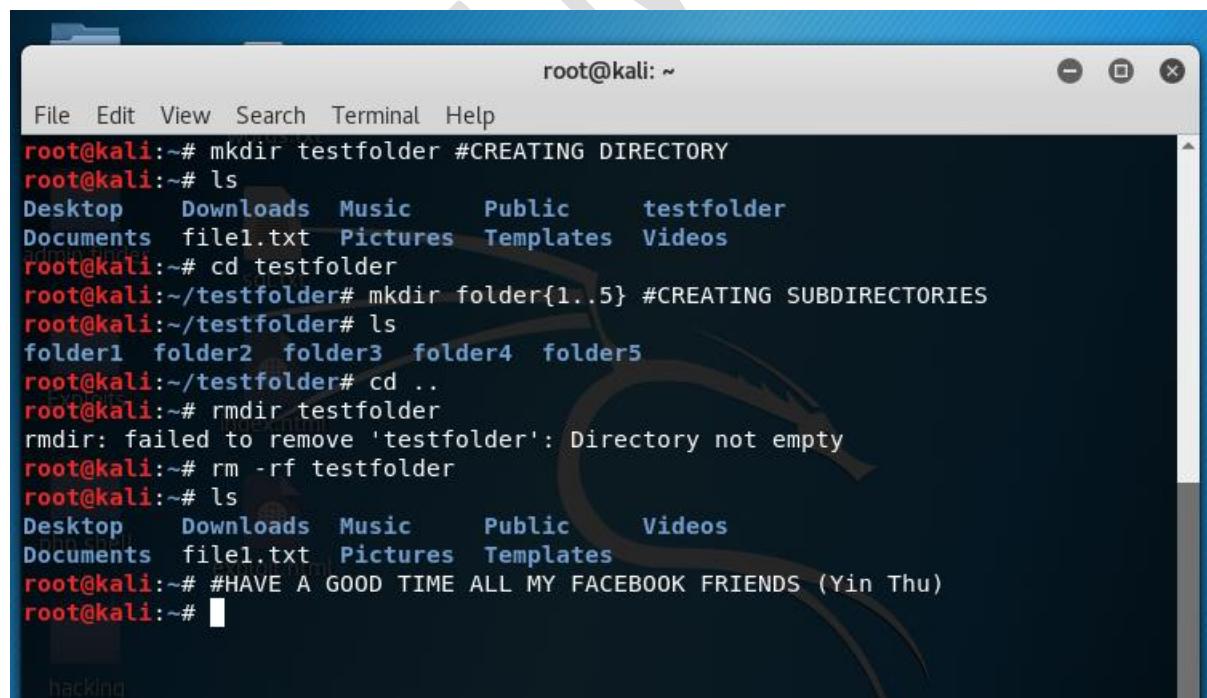
၆။ **directory** တစ်ခုကို ဖျက်ချင်ရင် **rmdir** သုံးရပါတယ်။ ဥပမာ **testfolder** အမည်ရတဲ့ **directory** ကို ဖျက်မယ်ဆိုပါစို့။ **rmdir dir1** လို့ ရိုက် **enter** ခေါက်ရုံပေါ့။ **rmdir testfolder**

၇။ ဒါပေမယ့်၊ ဖျက်ချင်တဲ့ **directory** ထဲမှာ **file** တွေ **sub directory** တွေ ရှိနေရင် ဖျက်ဖို့ မလွယ်ပါဘူး။ အဲဒီအခါ **rm -r -f** နဲ့ ဖျက်နိုင်ပါတယ်။ **rm -rf** လို့လည်း တွဲသုံးနိုင်ပါတယ်။ ဥပမာ **testfolder** အမည်ရတဲ့ **directory** ကို ဖျက်မယ်ဆိုပါစို့။ **rm -rf dir1** လို့ ရိုက် **enter** ခေါက်ရုံပေါ့။ **rm -r -f testfolder** သို့မဟုတ် **rm -rf testfolder**

လွယ်လွယ်လေးပါနော်။ **directory** နဲ့ သက်ဆိုင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေး။

Yin Thu

(14-5-2018)



The screenshot shows a terminal window titled "root@kali: ~". The terminal displays the following command sequence:

```
root@kali:~# mkdir testfolder #CREATING DIRECTORY
root@kali:~# ls
Desktop Downloads Music Public testfolder
Documents file1.txt Pictures Templates Videos
root@kali:~# cd testfolder
root@kali:~/testfolder# mkdir folder{1..5} #CREATING SUBDIRECTORIES
root@kali:~/testfolder# ls
folder1 folder2 folder3 folder4 folder5
root@kali:~/testfolder# cd ..
root@kali:~# rmdir testfolder
rmdir: failed to remove 'testfolder': Directory not empty
root@kali:~# rm -rf testfolder
root@kali:~# ls
Desktop Downloads Music Public Videos
Documents file1.txt Pictures Templates
root@kali:~# #HAVE A GOOD TIME ALL MY FACEBOOK FRIENDS (Yin Thu)
root@kali:~#
```

date and time

ခုတစ်ခါ linux ရဲ terminal ကနေ date and time သတ်မှတ်ပေးတဲ့ အပိုင်းကို ကြည့်ရအောင်။ server တွေကို run ရာမှာ date and time မှန်ကန်နေဖို့ အရေးကြီးပါတယ်။ ဒါကြောင့် date and time ဟာ လွယ်ကူပေမယ့် အရေးကြီးတဲ့ အပိုင်းတစ်ခုပေါ့။ စလိုက်ရအောင်။

၁။ **system** ရဲရက်စွဲနဲ့ အချိန်ကို သိချင်ရင် **date** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ **date** လို့ ရှိက် enter ခေါက်ပေါ့ **date** ဒီအခါ date and time ကို ပြသပေးပါလိမ့်မယ်။

၂။ ရက်စွဲ သတ်မှတ်ဖို့ အတွက် **date** ဆိုတဲ့ **command** နဲ့အတူ **+%Y%m%d -s** ဆိုတဲ့ **option** ကို တွဲသုံးရပါတယ်။ ဥပမာ **date** ကို ၂၀၁၈၊ မေလ၊ ၁၄ ရက်နေ့အဖြစ် သတ်မှတ်မယ် ဆိုပါစွဲ။ **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။ **date +%Y%m%d -s "20180514"**

ဒီနေရာမှာ **+%Y%m%d** သည် **year, month, date** ဆိုတဲ့ **format** ကို သုံးမယ်လို့ အမိပါယ် ရပါတယ်။ **-s** ကတော့ **set** သတ်မှတ်မယ်ပေါ့။

၃။ အချိန်ကို သတ်မှတ်ဖို့ အတွက် **date** ဆိုတဲ့ **command** ကို **+%T%p -s** ဆိုတဲ့ **option** နဲ့တွဲသုံးရပါတယ်။ ဥပမာ **time** ကို ညနေ ၅ နာရီ ၃၀ မိန့် သတ်မှတ်မယ် ဆိုပါစွဲ။ **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။ **date +%T%p -s "05:30:00PM"**

ဒီနေရာမှာ **+%T** သည် အချိန်ကို ကိုယ်စားပြုပါတယ်။ သူ့ရဲ့ **format** က **hour, minute, second** ပါ။ **%p** က **AM/PM** ကို ကိုယ်စားပြုပါတယ်။

၄။ အထက်ပါ နည်းလမ်းတွေက **system** ရဲ **date and time** ကို သတ်မှတ်တာပါ။ စက်ရဲ့ တကယ့် **hardware clock** ဆိုတာ ရှိပါတယ်။ သူက စက်ပိတ်သွားရင်တောင် **mother board** မှာ ပါရှိတဲ့ **battery** အားနဲ့ အလုပ်လုပ်နေပါတယ်။

ဒါကြောင့် **system** ရဲ **date and time** ကို သတ်မှတ်ပြီးသွားရင် အဲဒါကို **hardware clock** မှာပါ ထပ်တူကျဖို့ ထပ်မံ ပြုလုပ်ပေးရပါမယ်။ အသုံးပြုရမယ့် **command syntax** ကတော့ အောက်ပါ အတိုင်းပါ။ **hwclock -w** သို့မဟုတ် **hwclock --systohc**

ပြီးရင် **hardware clock** ကို ပြန်စစ်ကြည့်ပါ။ **hardware clock** ကို ပြန်ကြည့်နိုင်တဲ့ **command** က **hwclock --show** ပါ။ **hwclock --show**

၅။ **Time Zone** ကို သတ်မှတ်ဖို့ ကျနော်တို့ နိုင်ငံ အတွက်၊ **Asia/Yangon** ကို ရွေးပေးရပါ မယ်။ **command** ကတော့ အောက်ပါအတိုင်းပါ။

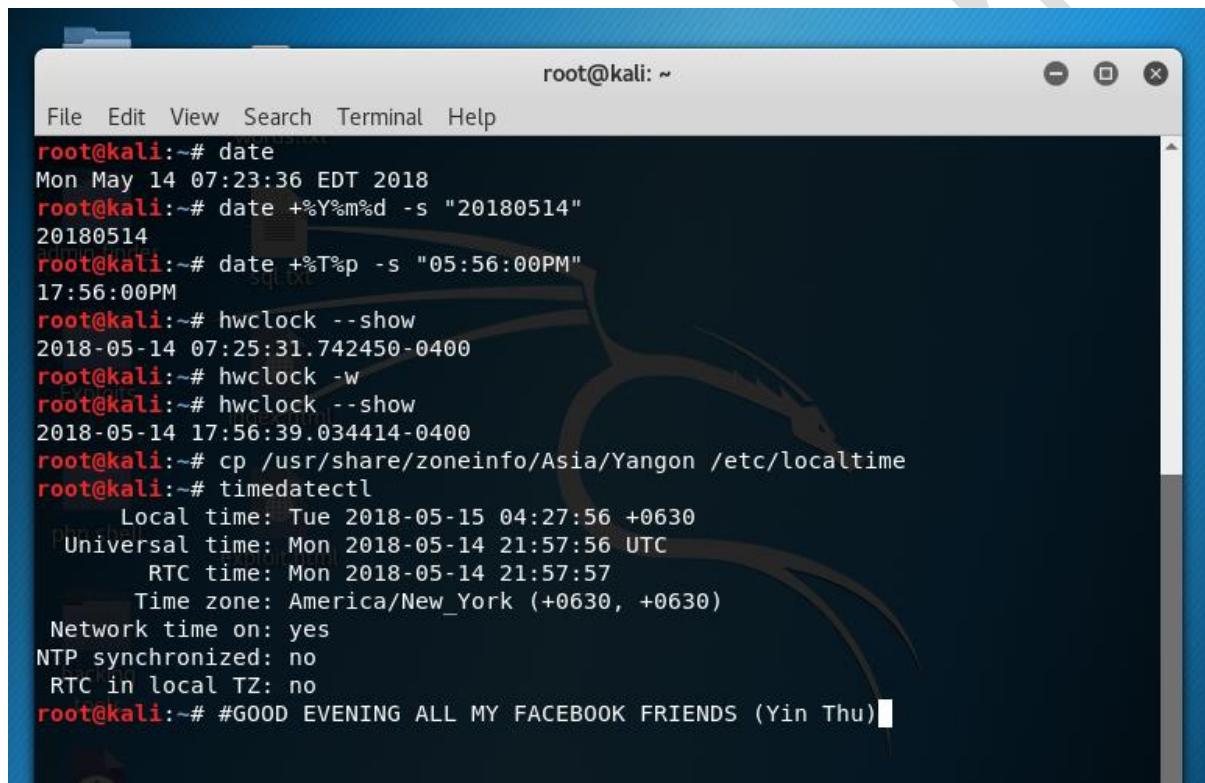
cp /usr/share/zoneinfo/Asia/Yangon /etc/localtime

ကဲ ! ခုလောက်ဆို date and time နဲ့ပတ်သက်လို့ ရှင်းလင်းသွားပြီလို့ ယူဆပါတယ်။
ခုလက်ရှိ update ဖြစ်တဲ့ linux os တွေမှာ **timedatectl** ဆိုတဲ့ **command** ကို အသုံးပြုပြီး **date and time** ကို သတ်မှတ်နိုင်ပါသေးတယ်။ သူက သုံးရတာ ပိုလွှယ်ကူလာတာပေါ့နော်။
timedatectl ဆိုတဲ့ **command** ရဲ့ အသုံးပြုပုံတွေကို နောက်ပိုင်း သင်ခန်းစာမှာ ထည့်သွင်းပေးသွားပါမယ်။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(14-5-2018)



```
File Edit View Search Terminal Help
root@kali:~# date
Mon May 14 07:23:36 EDT 2018
root@kali:~# date +%Y%m%d -s "20180514"
20180514
root@kali:~# date +%T%p -s "05:56:00PM"
17:56:00PM
root@kali:~# hwclock --show
2018-05-14 07:25:31.742450-0400
root@kali:~# hwclock -w
root@kali:~# hwclock --show
2018-05-14 17:56:39.034414-0400
root@kali:~# cp /usr/share/zoneinfo/Asia/Yangon /etc/localtime
root@kali:~# timedatectl
    Local time: Tue 2018-05-15 04:27:56 +0630
    Universal time: Mon 2018-05-14 21:57:56 UTC
        RTC time: Mon 2018-05-14 21:57:57
        Time zone: America/New_York (+0630, +0630)
    Network time on: yes
  NTP synchronized: no
    RTC in local TZ: no
root@kali:~# #GOOD EVENING ALL MY FACEBOOK FRIENDS (Yin Thu)
```

linux ကို GUI (Graphical User Interface) အပါအဝင် Application နဲ့အခြေခံ အသုံးပြုပုံတွေကို လေ့လာချင်တယ်ဆိုရင် ကိုအိမောင် (Fairway Technology) ရေးတဲ့ UBUNTU အပေါ် အခြေပြု ထားတဲ့ သင့်အတွက် linux စာအုပ်ကို အဆိုပြုပါရေး။ အောက်ပါ Link ကနေ download လုပ် ရယူ နိုင်ပါတယ်။

<https://eimaung.com/ubuntu-for-you/>

ကျေနော် page မှာကတော့ CLI (Command Line Interface) ကို ဦးစားပေးပြီး redhat, centos, fedora တွေကို အခြေပြုပြီး၊ linux system administration နဲ့ networking ပိုင်းကို ဖြည့်ဖြည့်ချင်း တင်ပေးသွားမှာပါ။

online မှာ linux system administration အကြောင်း စာအုပ်ကို လွတ်လပ်စွာ လေ့လာနိုင်ဖို့ 2017 တူန်းက ဖြန့်ထားပြီး ဖြစ်ပေမယ့်၊ အဲအတဲက၊ လိုအပ်တာတွေကို ထပ်မံတည်းဖြတ်၊ ထပ်တိုးပြီး တင်ပေးသွားမှာပါ။ စာအုပ်ဟောင်းကို အောက်ပါ Link ကနေ ရယူနိုင်ပါတယ်။

http://www.mediafire.com/.../Linux_System_Administration_%28c...

ကျေနှုံး tutorial တွေကတော့ GUI ပါမှာ မဟုတ်ပါဘူး။ linux terminal နဲ့ command တွေနဲ့သာ အမှန်း နှစ်ပါးသွားရမှာပါ။

page နဲ့ group တည်ထောင်ရခြင်း စာအုပ် ဖြန့်ရခြင်း ရည်ရွယ်ချက်က သင်တန်းမတက်နိုင်တဲ့ နယ်ကလေ့လာသူတွေ၊ ငွေကြေးမပြည့်စုံသေးသူတွေ၊ အဆင်ပြုဖို့ မိမိ သိသလောက် ပြန်လည် မျှဝေရုံပါ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(15-5-2018)



copy ကူးကြမယ်

linux machine ပေါ်မှာ **file** တွေ၊ **folder/directory** တွေကို တစ်နေရာ ဆီ **copy** ကူးတဲ့ အကြောင်း လေ့လာကြည့်ကြစို့။ စလိုက်ရအောင်များ.....!

terminal ကို ဖွင့်ပါ။ **cd /root/Desktop** လို့ ရိုက် **enter** ခေါက်ပါ။ ဒါဆို **root account** ခဲ့
Desktop ပေါ်ရောက်သွားပြီပေါ့။

၁။ အရင်ဆုံး စမ်းသပ်ဖို့ အတွက် **touch** command ကို သုံးပြီး **file** လေး ၅ ခု ခဲ့
ဆောက်ကြည့်ထားကြတာပေါ့။ **touch file{1..5}** ဆောက်ပြီးရင် **ls** နဲ့ ပြန်ကြည့်။

၂။ ပြီးရင် **mkdir** command ကို သုံးပြီး **directory** လေး ၃ ခုလောက် ဆောက်ရအောင်။
mkdir dir{1..3} ဆောက်ပြီးရင် **ls** နဲ့ ပြန်ကြည့်။

၃။ **file** တွေ **dir** တွေကို **copy** ကူးချင်ရင် **cp** ဆိုတဲ့ **command** ကို သုံးပါတယ်။
ဥပမာ **/root/Desktop** ပေါက **file1** ကို **/root/Documents** ဆီ **copy** ကူးချင်တယ် ဆိုပါစို့။
အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါပဲ။

cp /root/Desktop/file1 /root/Documents

ဒီနေရာမှာ သိထားရမှာက၊ **cp** ခဲ့ ရှေ့က၊ **/root/Desktop/file1** ဆိုတာကို **source**
လို့ခေါ်ပြီး **/root/Documents** ကို **destination** လို့ ခေါ်ပါတယ်။

ပြီးရင် **cd /root/documents** လို့ ရိုက် **enter** ခေါက်ပြီး **Document** ဆိုတဲ့ **directory**
ဆီသွားပါ။ **ls** နဲ့ စစ်ဆေးကြည့်ပါ။ **file1** ဆိုတာလေး ရောက်နေပါလိမ့်မယ်။

၄။ **file** အများကြီးကို **command** တစ်ကြောင်းတည်းနဲ့ **copy** ကူးလိုလည်း ရပါတယ်။
အဲဒီလို **copy** ကူးချင်ရင် **source** နေရာမှာ **file name** တွေကို တန်းစီ ထည့်ပြီး **destination**
နေရာမှာ **location** ကို တစ်ကြိမ်ပဲ ထည့်ပေးရင် ရပါပြီ။

ဥပမာ **/root/Desktop** အောက်က၊ **file1**, **file2** နဲ့ **file3** ကို **/root/Documents** ဆီ **copy**
ကူးချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါ။

cp /root/Desktop/file1 /root/Desktop/file2 /root/Desktop/file3 /root/Documents

ပြီးရင် **cd /root/Documents** လို့ ရိုက် **enter** ခေါက်ပြီး **Document** ဆိုတဲ့ **directory**
ဆီသွားပါ။ **ls** နဲ့ စစ်ဆေးကြည့်ပါ။ **file1**, **file2**, **file3** ဆိုတာလေးတွေ ရောက်နေပါလိမ့်မယ်။

၅။ **file** ကို **copy** လည်းကူးမယ်။ **file** အမည် **name** လည်း ပြောင်းမယ်ဆိုရင်
destination နေရာခဲ့နောက်မှာ **new name** ကို ပေးရပါမယ်။

ဥပမာ **root directory** အောက်က၊ **Desktop** အောက်က၊ **file5** ဆိုတာလေးကို
/root/Documents ဆီ **555** ဆိုတဲ့ အမည်အဖြစ်၊ **copy** ကူးချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့်
command ကတော့ အောက်ပါအတိုင်းပါ။

cp /root/Desktop/file5 /root/Documents/555

ပြီးရင် **cd /root/Document** လို့ ရှိက် **enter** ခေါက်ပြီး **Documents** ဆိုတဲ့ **directory** ဆိုသွားပါ။ **ls** နဲ့ စစ်ဆေးကြည့်ပါ။ **555** ဆိုတာလေး ရောက်နေပါလိမ့်မယ်။

၆။ **directory** တွေကို **copy** ကူးမယ်ဆိုရင်လည်း **file** တွေကို **copy** ကူးတဲ့ နည်းအတိုင်း ကူးနိုင်ပါတယ်။ ဥပမာ **root directory** အောက်က၊ **Desktop** အောက်က၊ **dir3** ဆိုတာလေးကို **/root/Documents** ဆို **copy** ကူးချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါ။ **cp /root/Desktop/dir3 /root/Documents**

ပြီးရင် **cd /root/Documents** လို့ ရှိက် **enter** ခေါက်ပြီး **Documents** ဆိုတဲ့ **directory** ဆိုသွားပါ။ **ls** နဲ့ စစ်ဆေးကြည့်ပါ။ **dir3** ဆိုတာလေး ရောက်နေပါလိမ့်မယ်။

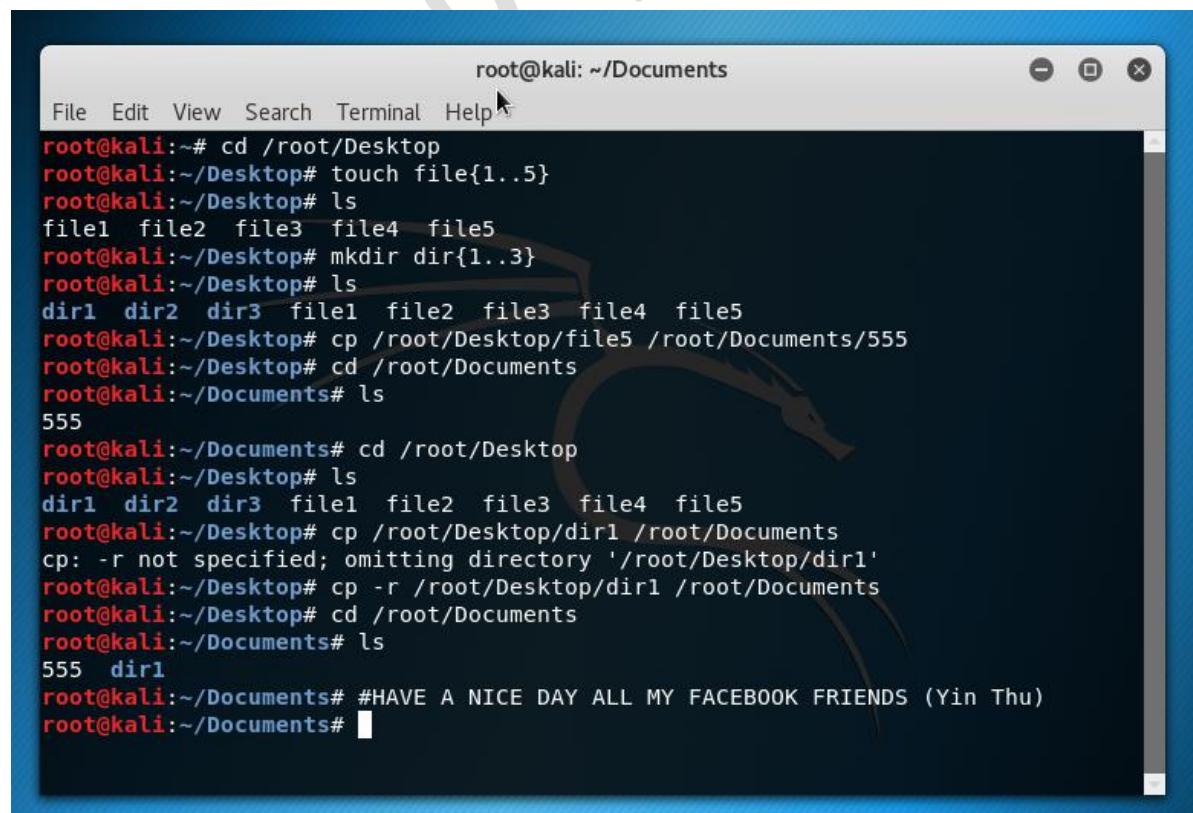
၇။ တစ်ခု သိထားရမှာက **file** တွေ ရှိနေတဲ့ **directory** ကို ကူးမယ်ဆို၊ ကူးလို့ မရပဲ၊ **cp : omitting directory** ဆိုတဲ့ **error** ပေါ်လေ့ရှိပါတယ်။

ဒါကြောင့် **file** တွေ ရှိတဲ့ **directory** ကို **copy** ကူးချင်တဲ့ အခါ -**r** ဆိုတဲ့ **option** လေးခံပေး ရပါတယ်။ အောက်ပါအတိုင်းပေါ့။ **cp -r /root/Desktop/dir1 /root/Documents**

လွှာယ်ပါတယ်နော်။ **copy** ကူးခြင်းနှင့် သက်ဆိုင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(15-5-2018)



The screenshot shows a terminal window titled "root@kali: ~/Documents". The terminal history is as follows:

```
root@kali:~# cd /root/Desktop
root@kali:~/Desktop# touch file{1..5}
root@kali:~/Desktop# ls
file1 file2 file3 file4 file5
root@kali:~/Desktop# mkdir dir{1..3}
root@kali:~/Desktop# ls
dir1 dir2 dir3 file1 file2 file3 file4 file5
root@kali:~/Desktop# cp /root/Desktop/file5 /root/Documents/555
root@kali:~/Desktop# cd /root/Documents
root@kali:~/Documents# ls
555
root@kali:~/Documents# cd /root/Desktop
root@kali:~/Desktop# ls
dir1 dir2 dir3 file1 file2 file3 file4 file5
root@kali:~/Desktop# cp /root/Desktop/dir1 /root/Documents
cp: -r not specified; omitting directory '/root/Desktop/dir1'
root@kali:~/Desktop# cp -r /root/Desktop/dir1 /root/Documents
root@kali:~/Desktop# cd /root/Documents
root@kali:~/Documents# ls
555 dir1
root@kali:~/Documents# #HAVE A NICE DAY ALL MY FACEBOOK FRIENDS (Yin Thu)
root@kali:~/Documents#
```

move and rename files and directories

linux ပေါ်မှာ **copy** ကူးတဲ့အကြောင်း ပြီးသွားပြီဆိုတော့ ဒီတခါ **cut** လုပ်တဲ့အကြောင်း **file** တွေ **directory** တွေကို အမည်ပြောင်းတဲ့အကြောင်း ဆက်သွားရအောင်။ **GUI** မှာ **cut** လုပ်တယ်လို့ ခေါ်ပြီး၊ linux ခဲ့ **command line** မှာတော့ ဒါကို **move** လုပ်တယ်လို့ ခေါ်ပါတယ်။ **terminal** ကို ဖွင့်ပါ။ **cd /root/Desktop** လို့ ရှိက် **enter** ခေါက်ပါ။ ဒါဆို **root account** ခဲ့ **Desktop** ပေါ်ရောက်သွားပြီပေါ့။

၁။ အရင်ဆုံး စမ်းသပ်ဖို့ အတွက် **touch command** ကို သုံးပြီး **file** လေး ၅ ခု ခေါက်ကြည့်ထားကြတာပေါ့။ **touch file{1..5}** ခေါက်ပြီးရင် **ls** နဲ့ ပြန်ကြည့်။

၂။ ပြီးရင် **mkdir command** ကို သုံးပြီး **directory** လေး ၃ ခုလောက် ခေါက်ရအောင်။ **mkdir dir{1..3}** ခေါက်ပြီးရင် **ls** နဲ့ ပြန်ကြည့်။

၃။ **file** တစ်ခု ကို **move** လုပ်မယ်ဆိုရင် **mv** ဆိုတဲ့ **command** ကို သုံးပါတယ်။ အမည်ပြောင်းမယ်ဆိုရင်လည်း အဲဒီ **mv** ကိုပဲ သုံးရပါတယ်။ ဥပမာ **/root/Desktop** အောက်က **file1** ဆိုတာလေးကို **/root/Documents** ဆို **move** လုပ်ချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါ။ **mv /root/Desktop/file1 /root/document**

ဒီနေရာမှာ သိထားရမှာကာ **mv** ခဲ့ ရှေ့ကာ **/root/Desktop/file1** ဆိုတာကို **source** လို့ ခေါ်ပြီး၊ **/root/Documents** ကို **destination** လို့ ခေါ်ပါတယ် ဆိုတာပါ။

၄။ **file** အများကြီးကို **command** တစ်ကြောင်းတည်းနဲ့ **move** လုပ်လို့လည်း ရပါတယ်။ **source** နေရာမှာ **file name** တွေကို တန်းစီ ထည့်ပြီး၊ **destination** နေရာမှာ **location** ကို တစ်ကြိမ်ပဲ ထည့်ပေးရင် ရပြီ။

mv /root/Desktop/file1 /root/Desktop/file2 /root/Desktop/file3 /root/Documents

၅။ **directory** ကို **move** လုပ်မယ်ဆိုရင်လည်း **file** တွေကို **move** လုပ်တဲ့ နည်းအတိုင်း ပါပဲ။ ဥပမာ **/root/Desktop** မှာ ရှိတဲ့ **dir1** ဆိုတဲ့ **directory** ကို **root directory** အောက်က **Documents** ဆိုတဲ့ **directory** အောက်ကို ရွှေ့မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါပဲ။ **mv /root/Desktop/dir1 /root/Documents**

၆။ **file** ကို အမည် **name** ပြောင်းမယ်ဆိုရင်လည်း **mv** ကိုပဲ သုံးပါတယ်။ **destination** နေရာခဲ့ နောက်မှာ **new name** ထည့်ပေးရတယ်။ ဥပမာ **current location** မှာ ရှိတဲ့ **file1** အမည်ရတဲ့ **file** ကို **hello** လို့ပြောင်းမယ်ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့အောက်ပါ အတိုင်းပါ။

mv file1 hello

ဥပမာ `/root/Desktop` အောက်မှာ ရှိတဲ့ **file2** အမည်ရတဲ့ **file** ကို `/root/Documents` ဆို **world** ဆိုတဲ့ အမည်သစ်နဲ့ ရွှေ့မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါ။ **`mv /root/Desktop/file2 /root/Documents/world`**

၂။ **directory** ကို အမည် **name** ပြောင်းမယ်ဆိုရင်လည်း **`mv`** ကိုပဲ သုံးပါတယ်။ **destination** နေရာရဲ့နောက်မှာ **new name** ထည့်ပေးရတယ်။

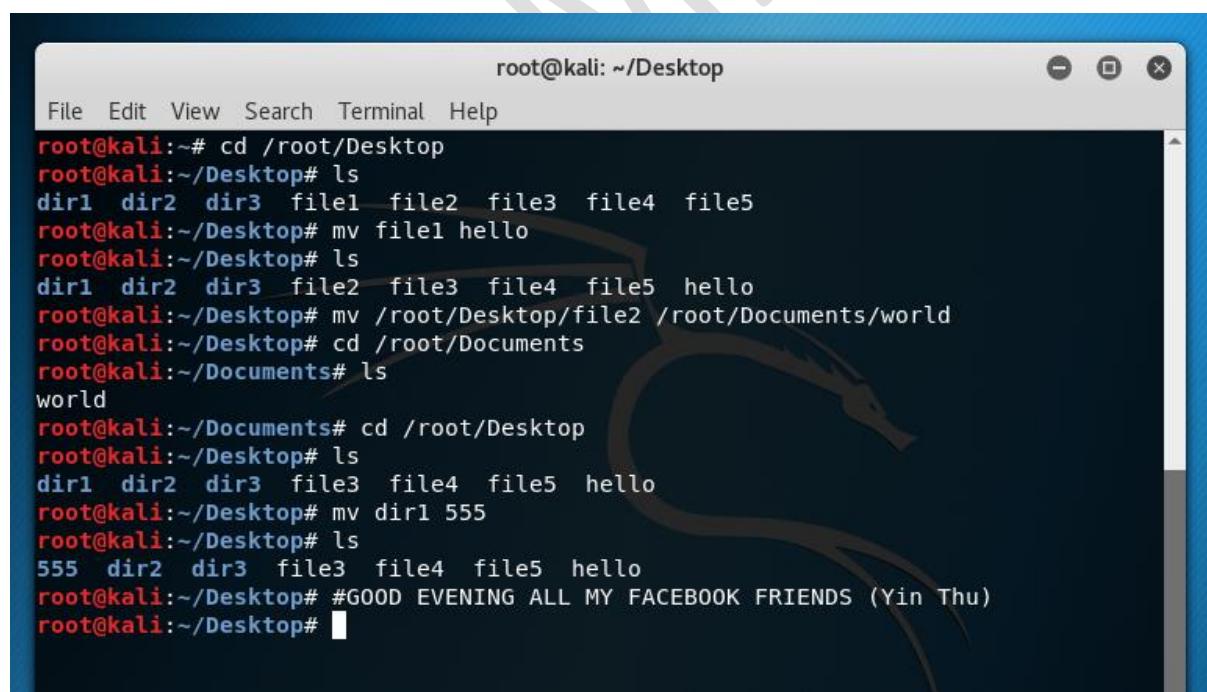
ဥပမာ **current location** မှာ ရှိတဲ့ **dir1** အမည်ရတဲ့ **directory** ကို **555** လို့ ပြောင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါ။ **`mv dir1 555`**

ဥပမာ `/root/Desktop` အောက်မှာ ရှိတဲ့ **dir2** အမည်ရတဲ့ **directory** ကို `/root/Documents` ဆို **333** ဆိုတဲ့ အမည်သစ်နဲ့ ရွှေ့မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါ။ **`mv /root/Desktop/dir2 /root/Documents/333`**

ကဲ ! ရှင်းရှင်းလေးပါပဲနော်။ file တွေ directory တွေကို move လုပ်ခြင်း၊ အမည်ပြောင်းခြင်းနဲ့ သက်ဆိုင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(15-5-2018)



```
root@kali:~/Desktop
File Edit View Search Terminal Help
root@kali:~# cd /root/Desktop
root@kali:~/Desktop# ls
dir1 dir2 dir3 file1 file2 file3 file4 file5
root@kali:~/Desktop# mv file1 hello
root@kali:~/Desktop# ls
dir1 dir2 dir3 file2 file4 file5 hello
root@kali:~/Desktop# mv /root/Desktop/file2 /root/Documents/world
root@kali:~/Desktop# cd /root/Documents
root@kali:~/Documents# ls
world
root@kali:~/Documents# cd /root/Desktop
root@kali:~/Desktop# ls
dir1 dir2 dir3 file3 file4 file5 hello
root@kali:~/Desktop# mv dir1 555
root@kali:~/Desktop# ls
555 dir2 dir3 file3 file4 file5 hello
root@kali:~/Desktop# #GOOD EVENING ALL MY FACEBOOK FRIENDS (Yin Thu)
root@kali:~/Desktop#
```

command တစ်ခုနှင့် ဆိုင်သော အကူအညီများကို ကြည့်ရခြင်း

linux command တွေရဲ့ အသုံးပြုပြုနည်း၊ ဘယ် **option** ကို ဘယ်လိုအခါမှာ တွဲဖက် ပေးရမယ်ဆိုတာ သိဖို့လိုပါတယ်။ အလွတ် မရတောင်မှ **help file** ကနေ ခေါ်ကြည့်တတ်ရင် အဆင်ပြေပါတယ်။ ဒီတော့ **command** တစ်ခုရဲ့ **manual**, **help file** တွေ၊ **info** တွေ၊ အချက် အလက်တွေကို ဘယ်နေရာမှာ ဝင်ရောက် ဖတ်ရှုရမယ်၊ စတာတွေကို လေ့လာကြည့်ရအောင်။

၁။ **echo \$PATH** >> command တွေရဲ့ တည်နေရာ လမ်းကြောင်းကို သိချင်ရင် **echo \$PATH** ကို သုံးပါတယ်။ **terminal** မှာ **echo \$PATH** လို့ ရှိက်၊ **enter** ခေါက်လိုက်တာနဲ့ **commands** တွေ တည်ရှိရာ လမ်းကြောင်းများကို ပြသပေးပါလိမ့်မယ်။ **echo \$PATH**

၂။ **which** >> **command** တစ်ခုရဲ့ တည်နေရာ လမ်းကြောင်းကို သိချင်ရင် **which** ကို သုံးနိုင်ပါတယ်။

ဥပမာ၊ **cp** ဆိုတဲ့ **command** ရဲ့ တည်နေရာကို သိချင်တယ် ဆိုပါစို့။ **which cp** ပေါ့။

which cp

၃။ **type** >> ဒါ့အပြင် **command** တစ်ခုချင်းရဲ့ တည်နေရာ လမ်းကြောင်းကို သိချင်ရင် **type** ကို သုံးနိုင်ပါသေးတယ်။

ဥပမာ၊ **mv** ဆိုတဲ့ **command** ရဲ့ တည်နေရာကို သိချင်တယ် ဆိုပါစို့။

type mv ပေါ့။ **type mv**

၄။ **--help** >> **command** တစ်ခုနဲ့ ဆိုင်တဲ့ အကူအညီတွေကို သိချင်ရင် **command** နောက်ကနေ **--help** ရှိက်ပြီး ကြည့်နိုင်ပါတယ်။ ဥပမာ၊ **mv** ဆိုတဲ့ **command** နဲ့ ဆိုင်တဲ့ အကူအညီကို ကြည့်မယ် ဆိုပါစို့။ **mv --help** ပေါ့။ **mv --help**

၅။ **man** >> **command** တစ်ခုရဲ့ အသုံးပြုပဲ **manual** ကို သိချင်ရင် **man** ဆိုတဲ့ **command** ကို သုံးပါတယ်။ **manual page** ကို ဖော်ပြုပေးမှာ ဖြစ်ပြီး သက်ဆိုင်တဲ့ **command** အရဲ့ သုံးပုံသုံးနည်းတွေကို ရှင်းပြထားတာ တွေ့ရမှာပါ။ **man page** က ထွက်ဖို့ အတွက် **q** ကို နှိပ်ရပါတယ်။

ဥပမာ၊ **mv** ဆိုတဲ့ **command** ရဲ့ **manual** ကို ကြည့်ချင်တယ် ဆိုပါစို့။ **man mv** ပေါ့။ **man mv**

၆။ **pinfo** >> **pinfo** ဆိုတာကေတွဲ **man** နဲ့ အလားတူတဲ့ **command** တစ်ခုပါပဲ။ **command** တစ်ခုနဲ့ ဆိုင်တဲ့ **info** တွေကို ပြသပေးပါတယ်။ **pinfo cp**

ဒီ **pinfo command** က **red hat** နဲ့ **centos linux** မှာပဲ ပါလာပါတယ်။

၇။ **info** >> **command** နဲ့ ဆိုင်တဲ့ **info** ကို ကြည့်ချင်ရင် **info** ကို သုံးနိုင်ပါသေးတယ်။

info mv ။ ဒီ **info command** က **red hat** နဲ့ **centos linux** မှာပဲ ပါလာပါတယ်။

၈။ မိမိ သွင်းလိုက်တဲ့ **application** နဲ့ ဆိုင်တဲ့ အချက်အလက် **documentation** တွေကို
ကြည့်ချင်ရင်တော့ **/usr/share/doc** ဆိုတဲ့ လမ်းကြောင်းထဲမှာ ရှာဖတ်လိုပါတယ်။
ကဲ ! command တစ်ခုနဲ့ သက်ဆိုင်တဲ့ manual ကြည့်နည်း၊ အကူအညီ ခေါ်ပုံ၊ info ကြည့်နည်း၊
documentation များကို ဝင်ရောက် ဖတ်ရှုနည်းတွေကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(17-5-2018)



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
root@kali:~# which cp
/bin/cp
root@kali:~# type mv
mv is /bin/mv
root@kali:~# #Type cp --help and press Enter
root@kali:~# #Type man mv and press Enter, To quit from man page >> press q key
root@kali:~# #You can use info and pinfo commands in centos linux
root@kali:~# #Be Happy All My Facebook Friends (Yin Thu)
root@kali:~#
```

head, tail, cat, more, less

linux ခဲ့ terminal ထဲကနေ text file တွေ config file တွေ log file တွေ၊ text result တွေ စသည်ဖြင့် စာသားတွေကို ဖတ်ချင်တယ် ဆိုပါစို့။ ဘယ် command တွေကို အသုံးပြုလို့ ရမလဲ ကြည့်ကြရအောင်။ ဒီလို့ text file တွေ၊ result တွေကို ကြည့်မယ် ဆိုရင်

၁။ head

J။ tail

၃။ cat

၄။ more

၅။ less စတဲ့ command တွေ အထဲကမှ တစ်ခုကို သင့်တော်သလို အသုံးပြုပြီး ကြည့်ရပါတယ်။ ဥပမာ၊ **readme.txt** ဆိုတဲ့ file လေး ရှိတယ် ဆိုပါစို့။

၁။ **head** >> သူက **file** ထဲမှာ ပါရှိတဲ့ စာသားတွေထဲကမှ ထိပ်ဆုံးစာ ၁၀ ကြောင်းကို ဖော်ပြပေးပါတယ်။ ဥပမာ၊ **readme.txt** ဆိုတဲ့ **file** ထဲက ထိပ်ဆုံးစာ ၁၀ ကြောင်းကို ကြည့်ချင်တယ် ဆိုပါစို့။ **head readme.txt** လို့ ရှိက် ရပါမယ်။ **head readme.txt** ကြည့်ချင်တဲ့ **line** အရေအတွက်ကို **head -n** လို့ သုံးပြီး သတ်မှတ်ပေးလို့ရပါတယ်။ **n** နေရာမှာ ကြည့်လိုတဲ့ **line** အရေအတွက်ကို ထည့်ပေးရုံပါပဲ။ ဥပမာ၊ **readme.txt** ဆိုတဲ့ **file** ထဲက ထိပ်ဆုံးစာ ၃ ကြောင်းကို ကြည့်ချင်တယ် ဆိုပါစို့။ **head -3 readme.txt**

J။ **tail** >> သူက **file** ထဲမှာ ပါရှိတဲ့ စာသားတွေထဲကမှ အောက်ဆုံးစာ ၁၀ ကြောင်းကို ဖော်ပြပေးပါတယ်။ ဥပမာ၊ **readme.txt** ဆိုတဲ့ **file** ထဲက အောက်ဆုံးစာ ၁၀ ကြောင်းကို ကြည့်ချင်တယ် ဆိုပါစို့။ **tail readme.txt** လို့ ရှိက် ရပါမယ်။ **tail readme.txt** **tail** ကို **-f** ဆိုတဲ့ **option** နဲ့ တွဲပြီး **file** တစ်ခု သို့မဟုတ် **output result** တစ်ခုကို **live** ကြည့်လို့ ရပါသေးတယ်။ **tail -f readme.txt**

tail -f readme.txt လို့ ရှိက် **enter** ခေါက်မယ်။ တစ်ဖက်မှာ၊ နောက်ထပ် **terminal** တစ်ခု ထပ်ဖွင့်ပြီး **readme.txt** ကို **edit** လုပ်မယ်၊ စာကြောင်းတွေ ထပ်ဖြည့်မယ်ဆိုရင်၊ ထပ်ဖြည့်လိုက်တဲ့ စာကြောင်းတွေဟာ **tail -f** နဲ့ ဖွင့်ထားတဲ့ **terminal** ထဲမှာ အချိန်နဲ့တစ်ပြီးညီ သွားပြောင်း လဲနေမှာပါ။ ကြည့်ချင်တဲ့ **line** အရေအတွက်ကို **tail -n** လို့ သုံးပြီး သတ်မှတ်ပေးလို့ရပါတယ်။ **n** နေရာမှာ ကြည့်လိုတဲ့ **line** အရေအတွက်ကို ထည့်ပေးရုံပါပဲ။ ဥပမာ၊ **readme.txt** ဆိုတဲ့ **file** ထဲက အောက်ဆုံးစာ ၃ ကြောင်းကို ကြည့်ချင်တယ် ဆိုပါစို့။ **tail -5 readme.txt** လို့ ရှိက် ရပါမယ်။

tail -3 readme.txt

၃။ **cat** >> သူကတော့ **file** ထဲမှာ ပါတဲ့ စာကြောင်း အားလုံးကို ပြသပေးပါတယ်။ **terminal** ညာဘက်ဘေးက **scroll bar** ကို အသုံးပြုပြီး အထက်အောက် ရွှေ့ကြည့်နိုင်ပါတယ်။

ဥပမာ **readme.txt** ဆိုတဲ့ **file** ထဲက စာကြောင်း အားလုံးကို ကြည့်ချင်တယ် ဆိုပါစို့။ **cat readme.txt** လို့ရှိက် ရပါမယ်။ **cat readme.txt**

ငါ။ **more** >> သူက စာကြောင်းရောန်သလောက် ပြဿပေးပြီး အဆုံးထိ ဖတ်ချင်ရင် **enter key** ခေါက်ပြီး အောက်ကို ၁ ကြောင်းချင်း ဆင်းသွားနိုင်ပါတယ်။ အထက်အောက် ရွှေလို့ တော့မရပါဘူး။ ဥပမာ **readme.txt** ကို ကြည့်ချင်တယ် ဆိုပါစို့။ **more readme.txt** လို့ရှိက် ရပါမယ်။ **more readme.txt**

၅။ **less** >> သူကတော့ သုံးလို့ ကောင်းပါတယ်။ စာကြောင်းတွေကို ဖတ်နေရင်း **arrow key** အသုံးပြုပြီး အထက်အောက် ရွှေ့ကြည့်နိုင်ပါတယ်။ ဥပမာ **readme.txt** ကို ကြည့်ချင်တယ် ဆိုပါစို့။ **less readme.txt** လို့ရှိက် ရပါမယ်။ **less readme.txt** ဖတ်နေရင်းနဲ့ ထွက်ချင်ရင် **Esc** နှင့် **q** လို့ရှိပိုး ထွက်နိုင်ပါတယ်။

ဒါကတော့ text file တွေကို ဖတ်တာပေါ့နော်။ နောက်တစ်ခုက text result တွေကို ဖတ်တာပါ။

text result

text result ဆိုတာက **command** တစ်ခုကို **run** လိုက်ရင် **text output** တွေ ပြဿပေးတာကို ဆိုလိုတာပါ။ ဥပမာ **journalctl** (journalctl အကြောင်းကို နောက်ပိုင်းတွင် ရှင်းပါမည်) လို့ရှိက် ခေါက်လိုက်ရင် **system log** နဲ့ဆိုင်တဲ့ **text result** တွေ အများကြီး ထွက်လာမှာပါ။ **text result** ဆိုတာ ဒါကို ဆိုလိုတာပါ။ ဒီလို **text result** တွေကို ဖတ်ဖို့အတွက် **more** နဲ့ **less** ကို အသုံးများကြပါတယ်။ ဘယ်လို့ သုံးလဲဆိုတော့ **pipe** သက်တ ။ ခံပြီး သုံးကြပါတယ်။ အောက်ပါအတိုင်းပါ။ **journalctl | less**

ကဲ ! text file တွေ၊ text result ကို ကြည့်ပုံကြည့်နည်းကတော့ ဒီလောက်ပါပဲ။

command တွေကို ကာလံဒေသံအလိုက် သင့်တော်သလို အသုံးချိန်ပါစေး။

အားလုံး အဆင်ပြေပါစေး။

Yin Thu

(17-5-2018)

The screenshot shows a terminal window titled "root@kali: ~". The terminal displays the following command history:

```
root@kali:~# head -2 readme.txt
1.Yin Latt
2.Ei Chaw Po
root@kali:~# tail -2 readme.txt
2.Ei Chaw Po
3.Aye Thaung
root@kali:~# cat readme.txt
1.Yin Latt
2.Ei Chaw Po
3.Aye Thaung
root@kali:~# more readme.txt
1.Yin Latt
2.Ei Chaw Po
3.Aye Thaung
root@kali:~# less readme.txt
root@kali:~# journalctl | less
root@kali:~#
root@kali:~# #Have A Nice Time All My Facebook Friends (Yin Thu)
root@kali:~#
```

creating and editing text file in linux

linux မှာ server တွေ တင်ပြီး configuration လုပ်တဲ့ အခါ .cfg, .conf, .. စတဲ့ file တွေထဲ ဝင်ပြီး ပြင်ဆင်တာ၊ ရေးသားတာ ပြုလုပ်ရပါတယ်။ အဲတော့ ဒီတစ်ခါ linux ပေါ်မှာ စာရေးမယ်။ ပြင်မယ်။ text file တွေကို create လုပ်မယ်။ edit လုပ်မယ်ပေါ့။ အရင်ဆုံး စမ်းသပ်ဖို့ အတွက် **touch** command ကို အသုံးပြုပြီး **readme.txt** ဆိုတဲ့ file လေးကို ဆောက်လိုက်ပါ။

touch readme.txt ကဲ ! text editing နဲ့ ပတ်သက်တာလေး ပြောကြရအောင်။

text file တွေကို creating နဲ့ editing လုပ်နိုင်ဖို့ linux မှာ text editor တွေ အများကြီး ရှိပါတယ်။ အဲဒီထက်မှာ **vim** နဲ့ **gedit** ဆိုတဲ့ text editor နှစ်မျိုးက လူသုံးများပါတယ်။ **vim >> သူကတော့ command line အခြေပြု text editor ပါ။ terminal ကို ဖွင့်ပြီး text file တွေကို edit လုပ်ချင်တဲ့ အခါ vim ကို သုံးရပါတယ်။**

၁။ **vim** text editor ကို ဖွင့်ဖို့ သုံးရမယ့် **command** ကတော့ **vim** ပါ။ **vi** လို့လည်း သုံးလို့ရပါတယ်။ ဥပမာ **readme.txt** ကို **edit** လုပ်ချင်တယ် ဆိုပါမို့။ ဒါအတွက် ရိုက်ရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါပဲ။ **vim readme.txt**

၂။ **vim editor** ထဲ ရောက်သွားပြီဆိုရင် **edit** လုပ်နိုင်ဖို့ အတွက် keyboard ပေါ်က **a** သို့မဟုတ် **i** သို့မဟုတ် **insert key** ကို နှိပ်ပေးရပါမယ်။ ဒါဆိုရင် **edit mode** ထဲ ရောက်သွားမှာပါ။ ဒါမှ စာရေးလို့ ရပါမယ်။

၃။ ပြီးရင် ကိုယ် ရေးမှတ်ချင်တာတွေ ရေးမှတ်ပေါ့။ စာသားတွေကို mouse အသုံးပြုပြီး selection မှတ်နိုင်ပါတယ်။

1. Yin Latt

2. Ei Chaw Po

3. Aye Thaung လို့ ရိုက်ကြည့်ပါ။ အစမ်းပေါ့။

၄။ ကိစ္စပြီးလို့ **save** လုပ်ပြီး ထွက်မယ်ဆို keyboard ပေါ်က **Esc** ကို ရိုက်၊ **:wq** လို့ ထပ်ရိုက်ပြီး ထွက်ရပါမယ်။ **save and exit** လုပ်တဲ့ သဘောပေါ့။ **:wq** နဲ့ **save and exit** လုပ် ထွက်လို့ မရရင် **:wq!** နဲ့ ထွက်ရပါတယ်။ **force** သုံးပြီး ထွက်တဲ့ သဘောပါ။

၅။ **save** မလုပ်ပဲ ထွက်မယ်ဆို **Esc** ရိုက်၊ **:q** လို့ ရိုက်ပြီး ထွက်ရုံပါပဲ။ **:q** နဲ့ **exit** လုပ် ထွက်လို့ မရရင် **:q!** နဲ့ ထွက်ရပါတယ်။ **force** သုံးပြီး ထွက်တဲ့ သဘောပါ။

၆။ စာကြောင်းတွေကို စာလုံးတစ်လုံးချင်းစီ ဒါမှုမဟုတ် **line** အလိုက် **block** အလိုက် **selection** မှတ်ချင်ရင် **visual mode** ကို ပြောင်းရပါမယ်။ **visual mode** ကို ပြောင်းမယ်ဆို **Esc** ကို နှိပ်ပြီးမှ **v** ကို နှိပ်ရပါတယ်။ **visual mode** ၃ မျိုး ရှိပါတယ်။ တစ်ခု သိယားရမှာကာ၊ **mode** တစ်ခုကနေ တစ်ခုဆီ ပြောင်းတဲ့ အခါ **Esc key** နှိပ်ပြီးမှသာ ကူးပြောင်းနိုင်တယ် ဆိုတာပါပဲ။

✓ အသေးကို နှိပ်ရင် သာမန် **visual mode** ဆီ ရောက်ပါမယ်။ **shift + v** ကို နှိပ်ရင် **line** အလိုက် **selection** လုပ်နိုင်တဲ့ **visual line mode** ဆီ ရောက်ပါမယ်။ **ctrl + v** ကို နှိပ်ရင် **block** အလိုက် **selection** လုပ်နိုင်တဲ့ **visual block mode** ဆီ ရောက်ပါမယ်။ **selection** မှတ်ခြင်းကို **keyboard** ပေါ်က arrow key တွေနဲ့ ဆောင်ရွက်ရပါတယ်။

၇။ **visual mode** ထဲမှာ **selection** မှတ်ပြီး **copy** ကူးတာကို **yank** လို့ ခေါ်ပြီး **y** key ကို အသုံးပြုရပါတယ်။ **visual mode** ထဲမှာ **copy** ကူးထားတာတွေ နေရာချု **paste** လုပ်တာကို **put** လို့ ခေါ်ပြီး၊ **p** key သုံးပါတယ်။ **p** ကို နှိပ်လိုက်ရင် **cursor** ရှိတဲ့ နေရာမှာ ကူးထားတာတွေကို နေရာချုသွားမှာပါ။

၈။ **visual mode** ထဲမှာ **selection** မှတ်ပြီး **copy cut** လုပ်မယ်ဆို **d** key ကို အသုံးပြုရပါတယ်။ **visual mode** ထဲမှာ **cut** လုပ်ထားတာတွေ နေရာချု **paste** လုပ်တာကို **put** လို့ ခေါ်ပြီး၊ **p** key သုံးပါတယ်။ **p** ကို နှိပ်လိုက်ရင် **cursor** ရှိတဲ့ နေရာမှာ **cut** လုပ်ထားတဲ့ **text** တွေကို နေရာချုသွားမှာပါ။

၉။ **visual mode** ထဲမှာ စာကြောင်းတွေကို **line** အလိုက် **delete** လုပ်မယ်ဆို **d** key ကို သုံးရပါတယ်။ **d** ကို နှိပ်တာနဲ့ ၁ ကြောင်းချင်း ဖျက်သွားမှာပါ။ **x** ကို နှိပ်ရင် **normal delete** ပါ။ ပုံမှန်အတိုင်း ဖျက်သွားပါလိမ့်မယ်။

၁၀။ ဆောင်ရွက်ထားတာတွေကို **undo** လုပ်မယ်ဆို **u** key ကို နှိပ်၊ **redo** လုပ်မယ်ဆို **ctrl + r** ကို အသုံးပြုရပါမယ်။ vim text editor နဲ့ ပတ်သက်တဲ့ တစ်ခြားသုံးနိုင်တာတွေကို သိလိုင်တော့ **man vi** လို့ရိုက် **enter** ခေါက်ပြီး **manual page** ထဲမှာ လေ့လာနိုင်ပါတယ်။

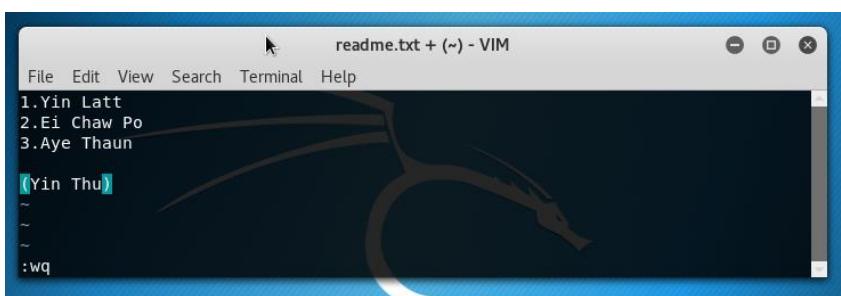
gedit >> သူကတော့ graphical interface နဲ့ text editor လေးပါ။ graphic ဖြစ်တာကြောင့် အသုံးပြုပုံက ရိုးရှင်း လွယ်ကူပါတယ်။ ဒါကြောင့် အကျယ်တဝ် ရှင်းပြမနေတော့ ပါဘူး။ ဥပမာ **readme.txt** ကို edit လုပ်ချင်တယ် ဆိုပါစို့။ ဒါအတွက် ရိုက်ရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါပဲ။ **gedit readme.txt** ဒီလိုရိုက် **enter** ခေါက်လိုက်ရင် **readme.txt** ဟာ **gedit application** လေးထဲမှာ ပွင့်လာပါလိမ့်မယ်။ သူရဲ့ graphical interface ကို လေ့လာပြီး edit လုပ်ရုံပေါ့။

ကဲ ! linux ပေါ်မှာ text file တွေကို creating and editing လုပ်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(18-5-2018)



creating user account

ဒီတစ်ခါ linux operating system မှာ user account တွေ တည်ဆောက်တဲ့အကြောင်းပြောရအောင်။ ကိုယ် အသုံးပြုနေတဲ့ account name ကို သိချင်ရင် terminal မှာ **whoami** လို့ရှိက် enter ခေါက်ပြီး သိနိုင်ပါတယ်။ **whoami** အဲဒါ အသာထားပြီး ရှေ့ဆက်သွားကြရအောင်။

#user_account

linux မှာ user account အမျိုးအစား ၃ မျိုးရှိပါတယ်။ အားဒါတွေကတော့၊

၁။ superuser

၂။ system user

၃။ normal user တို့ပါပဲ။

၁။ **superuser** ဆိုတာ **root account** ကို ပြောတာပါ။ **linux system** တစ်ခုလုံးမှာ လုပ်ပိုင်ခွင့် အရှိခံး account ပေါ့။ သူ၏ UID (User ID) က 0 ပါ။

၂။ **system user** ဆိုတာကတော့ **linux system** အတွက် လိုအပ်တဲ့ account တွေကို ဆိုလိုတာပါ။ operating system က သူဖော်တဲ့ account တွေပေါ့။

၃။ **normal user** ဆိုတာကတော့ ရှင်းပါတယ်။ အသစ် ထပ်မံတည်ဆောက်လိုက်တဲ့ account တွေပေါ့။ root account လို့ အလုံးစုံ လုပ်ပိုင်ခွင့် မရှိပါဘူး။

#user_id

ဒီနေရာမှာ တစ်ခု ပြောချင်တာက UID (User ID) ပါ။ account တည်ဆောက်စဉ်မှာ တွဲလျက်ပါလာတဲ့ ID ပေါ့။

၁။ **UID 0** ဟာ **root account** ခဲ့ **UID** ပါ။

၂။ **UID 1** ကနေ **999** ထိကို **system** က သုံးတဲ့ **UID** တွေအဖြစ် သတ်မှတ်ပါတယ်။ **os** က အသုံးပြုတဲ့ **account** တွေ အတွက်ပေါ့။

၃။ **UID 1000** ကနေ **60,000** ထိကို ထပ်မံ create လုပ်ထားတဲ့ **normal user account** တွေ အတွက် အသုံးပြုပါတယ်။

UID, GID သတ်မှတ်တဲ့နေရာမှာ နောက်ဆုံးဆောက်ခဲ့တဲ့ UID, GID (Group ID) ကို ကြည့်ပြီး တစ်တိုးသွားလေ့ရှိပါတယ်။ ဥပမာ ပေးရရင် **tom** ဆိုတဲ့ **user** ခဲ့ **UID** က 1,000 ဖြစ်တယ်။ အဲဒါကို **jerry** ဆိုတဲ့ အမည်နဲ့ user account တစ်ခု ထပ်လုပ်လိုက်တယ်။ **jerry** ဆိုတဲ့ **account** ခဲ့ **UID** သည် ဘယ်လောက် ဖြစ်မလဲ။ ရှိုးရိုးလေးပါ။ **1001** ဖြစ်သွားမှာပေါ့။

UID ခဲ့ default maximum value ကို **/etc/login.defs** မှာ ဝင်ရောက် ပြင်ဆင်လို့ ရပါတယ်။ ဘယ်လို့ ပြင်ရမလဲ ဆိုတော့ --> **vi /etc/login.defs** လို့ရှိက် enter ခေါက်။ **i** ကို နှိပ်

insert mode ကို ဝင်။ **UID_MAX 60000** နေရာမှာ **UID_MAX 1001** လိုပြင်။ **Esc** နှင့် **:wq** နဲ့ save and exit လုပ်။

အဲတော့ ဘာဖြစ်လဲ? normal user account နှစ်ခုထက် ပိုဆောက်လို့ မရတော့ဘူးပေါ့။ (normal user account တွေရဲ့ UID က 1000 ကနေ စတာမို့ 1001 ဆိုတော့ ၂ ခုပေါ့) ဒီသဘောပါ။ ဒါပေမယ့် အဲဒီ UID သတ်မှတ်ချက် range တွေဟာ linux os version ပေါ် မူတည်ပြီး ကဲ့
ပြားလေ့ရှိပါတယ်။ **centos 6** မှာ **normal user account** တွေ အတွက် **UID 500** ကနေ စပြီး **centos 7** မှာတော့ **UID 1,000** ကနေ စပါတယ်။

အထက်မှာ ဖော်ပြထားသော UID ဆိုင်ရာ သတ်မှတ်ချက်သည် centos 7 က အချက် အလက်တွေပါ။ သုံးထားတဲ့ photo ကတော့ centos 6 မှာ စမ်းထားတဲ့ ပုံပါ။ ဒါ့ကြောင့် tom ဆိုတဲ့ normal user account ရဲ့ UID က 500 ကနေ စနေတာပါ။ (Admin လည်း centos 7 မသွင်းရသေးလို့) :P

တဖန် account တစ်ခုရဲ့ UID ကို သိချင်ရင် **id** ဆိုတဲ့ command နဲ့ ကြည့်လို့ရပါတယ်။ ဥပမာ tom ဆိုတဲ့ account ရဲ့ UID ကို သိချင်တယ် ဆိုပါစို့။ **ID tom** လို့ ရိုက် enter ခေါက်။ **id tom** ဒါက user account နဲ့ သက်ဆိုင်တဲ့ UID အကြောင်းပါ။

#creating_account

user account တည်ဆောက်တဲ့ အကြောင်း ဆက်သွားလိုက်ကြစို့။

၁။ user account တစ်ခု တည်ဆောက်မယ်ဆို **useradd** ဆိုတဲ့ command ကို သုံးပါတယ်။ ဥပမာ tom ဆိုတဲ့ user account တည်ဆောက်မယ် ဆိုပါစို့။ **useradd tom** လို့ ရိုက် enter ခေါက်။

useradd tom

၂။ user account တစ်ခုကို password ပေးချင်ရင် **passwd** ဆိုတဲ့ command ကို သုံးပါတယ်။ ဥပမာ tom ဆိုတဲ့ account ကို password ပေးမယ် ဆိုပါစို့။ **passwd tom** လို့ ရိုက် enter ခေါက်။ **passwd tom**

၃။ user account တစ်ခုကို ဖျက်ချင်ရင်တော့ **userdel** ဆိုတဲ့ command ကို **-r** ဆိုတဲ့ option နဲ့ တွဲသုံးရပါတယ်။ ဥပမာ tom ဆိုတဲ့ user account ကို ဖျက်ချင်တယ် ဆိုပါစို့။ **userdel -r tom** လို့ ရိုက် enter ခေါက်။ **userdel -r tom**

တစ်ခု သိထားရမှာက user account တစ်ခု create လုပ်တာနဲ့ အမည်တူ group တစ်ခုပါ အလိုလို ရလာတယ်ဆိုတာပါပဲ။ ဥပမာ tom ဆိုတဲ့ user account ကို create လုပ်ရင် tom ဆိုတဲ့ အမည်နဲ့ group တစ်ခုပါ ဆောက်ပီးသား ဖြစ်သွားပါလိမ့်မယ်။

#user_info

user account တွေရဲ့ info တွေကို **/etc/passwd** (/ အောက်က **etc** ဆိုတဲ့ directory အောက်က **passwd** အမည်ရတဲ့ **file**) မှာ ဝင်ရောက် ကြည့်ရှုနိုင်ပါတယ်။

less /etc/passwd သိမဟုတ် **vim /etc/passwd**

/etc/passwd မှာ **user name, password, user id, group id, GECOS, home directory** နဲ့ **shell type** ဆိုတဲ့ အပိုင်း ၃ ပိုင်း -->

(1.username 2.password 3.UID 4.GID 5.GECOS 6.home dir 7.login shell) ကို colon

: နဲ့ခြားပြီး ဖော်ပြထားတာ တွေ့ရပါလိမ့်မယ်။ ရှင်းရရင် -->

၁။ **username** ဆိုတဲ့ နေရာမှာ ရှိတာသည် **username** ပေါ့။ ဥပမာ **root, tom,** စသည်ဖြင့် တွေ့ရပါမယ်။

၂။ **password** ဆိုတဲ့ နေရာမှာ ရှိတာသည် **password** ပေါ့။

၃။ **UID** ဆိုတဲ့ နေရာမှာ ရှိတာသည် **User ID** နံပါတ်ပေါ့။

၄။ **GID** ဆိုတဲ့ နေရာမှာ ရှိတာသည် အဲဒီ user ပါဝင်ထားတဲ့ group ရဲ့ **Group ID** ပေါ့။

၅။ ဒီနေရာမှာ **GECOS** ဆိုတာ account နဲ့ ပတ်သက်တဲ့ comment ထည့်သွင်းထားတဲ့ နေရာကို ဆိုလိုတာပါ။ GECOS ရဲ့အဓိပ္ပာယ်က **General Electric Comprehensive Operating System** ပါ။

၆။ **home directory** ဆိုတာကတော့ အဲဒီ user account ရဲ့ **ပင်မ directory** ကို ပြောတာပါ။

၇။ **shell type** ဆိုတာကတော့ အဲဒီ user အသုံးပြုတဲ့ **shell အမျိုးအစား** ကို ပြောတာပါ။

login ဝင်စရာ မလိုပဲ system ရဲ့ **လိုအပ်ချက်အရ တည်ဆောက်ထားတဲ့ account** တွေရဲ့ **shell နေရာမှာ /sbin/nologin** လို့ ပါလေ့ရှိတယ်။

ဒါအပြင် user နဲ့ ဆိုင်တဲ့ UID, GID နဲ့ ပါဝင်ထားတဲ့ group တွေကို **id** ဆိုတဲ့ command နဲ့ ကြည့်နိုင်ပါတယ်။

ဥပမာ **tom** ဆိုတဲ့ account ရဲ့ UID, GID, group တွေကို ကြည့်မယ် ဆိုပါစို့။ **ID tom** လို့ ရှိက် enter ခေါက်။ **ID tom**

user account အမျိုးအစား၊ တည်ဆောက်ပုံ၊ info ကြည့်ပုံတွေကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(18-5-2018)

```
[root@localhost ~]# whoami
root
[root@localhost ~]# useradd tom
[root@localhost ~]# passwd tom
Changing password for user tom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# id tom
uid=500(tom) gid=500(tom) groups=500(tom)
[root@localhost ~]# less /etc/passwd
[root@localhost ~]# #Good Evening All My Facebook Friends (Yin Thu)
[root@localhost ~]#
```

creating group

linux ပေါ်မှာ user account တွေ create လုပ်တာ သိပြီးတဲ့အနေက group တွေ အကြောင်း ပြောကြရအောင်။ group တွေကို ဘုရားကြောင့် ဖန်တီးရတာလဲဆိုတော့ group တစ်ခုထဲမှာ ပါဝင်တဲ့ user account တွေ အနေနဲ့ group owner က သတ်မှတ်ပေးထားတဲ့ policy အတိုင်း application, file, folder တွေကို မျှဝေ သုံးစွဲနိုင်ဖို့ပါ။

ဥပမာ teamone ဆိုတဲ့ group ထဲမှာ ပါဝင်တဲ့ member တွေသည် ဘယ် file တွေကို access လုပ်ခွင့် ရှိမယ်။ မပါဝင်တဲ့ account တွေကတော့ access လုပ်ခွင့် မရှိဘူး စသည်ဖြင့်ပေါ့။

#group_type

linux မှာ

၁။ **primary group** နဲ့

၂။ **supplementary group** ဆိုပြီး group အမျိုးအစား ၂ ခု ရှိပါတယ်။

၁။ **primary group** ဆိုတာကတော့ user account တစ်ခု create လုပ်စဉ်မှာ တွဲလျက်ပါလာတဲ့ default group ကို ဆိုလိုတာပါ။ ဥပမာ tom ဆိုတဲ့ user account ကို ဆောက်လိုက်ရင် tom ဆိုတဲ့ group လေးပါ တစ်ခါတည်း ဆောက်ပြီးသား ပါလာမှာပါ။ user account တစ်ခုဟာ ဘယ် group တွေထဲမှာ member အဖြစ် ပါဝင်နေသလဲ သိချင်ရင် **groups** ဆိုတဲ့ command နဲ့ ကြည့်နိုင်ပါတယ်။ ဥပမာ tom ဆိုတဲ့ user ဟာ ဘယ် group တွေရဲ့ member အဖြစ် ဝင်ထားသလဲ သိချင်တယ် ဆိုပါစို့။ **groups tom** လို့ ရိုက် enter ခေါက်။ **groups tom**

၂။ **supplementary group** ဆိုတာကတော့ user ကနေ ထပ်မံ တည်ဆောက်လိုက်တဲ့ group တွေကို ဆိုလိုတာပါ။

ကဲ ! ဟုတ်ပြီ group တွေကို တည်ဆောက်တဲ့အကြောင်း ဆက်ရ အောင်။

#creating_group

၁။ group တစ်ခု တည်ဆောက်မယ်ဆို **groupadd** ဆိုတဲ့ command ကို သုံးရပါတယ်။ ဥပမာ **teamone** ဆိုတဲ့ group တစ်ခု ဆောက်ချင်တယ် ဆိုပါစို့။ **groupadd teamone** လို့ ရိုက် enter ခေါက်။ **groupadd teamone**

၂။ group တစ်ခုကို ဖျက်ချင်ရင် **groupdel** ဆိုတဲ့ command ကို သုံးရပါတယ်။ ဥပမာ **teamone** ဆိုတဲ့ group ကို ဖျက်ချင်တယ် ဆိုပါစို့။ **groupdel teamone** လို့ ရိုက် enter ခေါက်။

groupdel teamone

#group_info

၁။ group တွေရဲ့ info တွေကို **/etc/group** မှာ ဝင်ကြည့်နိုင်ပါတယ်။ **less /etc/group** သို့မဟုတ် **vim /etc/group**

J|| ဘယ် group ထဲမှာ ဘယ် user တွေ ပါသလဲ သိချင်ရင် **/etc/gshadow** မှာ ဝင်ကြည့်နိုင်ပါတယ်။ **less /etc/gshadow** သို့မဟုတ် **vim /etc/gshadow** ကဲ ! group တစ်ခု create လုပ်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြောကြပါစေ။

Yin Thu

(19-5-2018)

```
root@localhost:~#
File Edit View Search Terminal Help
[root@localhost ~]# groupadd teamone
[root@localhost ~]# less /etc/group
[root@localhost ~]# less /etc/gshadow
[root@localhost ~]# groups tom
tom : tom
[root@localhost ~]# groupdel teamone
[root@localhost ~]# #Have A Good Time All My Friends (Yin Thu)
[root@localhost ~]#
```

user account နှင့် သက်ဆိုင်သော

linux မှာ user account တွေ တည်ဆောက်တာကို ပြောခဲ့ပြီးပါပြီ။ ဒီတစ်ခါ user account တွေကို လိုအပ်ချက်အရ ဘယ်လို modify လုပ်နိုင်မလဲဆိုတာ လေ့လာကြရအောင်။ user account တစ်ခုကို modify လုပ်ချင်တဲ့အခါ **usermod** ဆိုတဲ့ **command** ကို **option** တွေနဲ့ တွဲသုံးရပါတယ်။

၁။ **usermod -l** >> မိမိ ရဲ့ user name ကို change ချင်တဲ့အခါ သုံးပါတယ်။ **usermod -l oldname newname** ဥပမာ **tom** ဆိုတဲ့ **user name** ကို **jerry** လို့ ပြောင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **usermod -l tom jerry**

J|| **usermod -u** >> user account တစ်ခုရဲ့ **UID** (User ID) ကို ပြောင်းလဲချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ user ရဲ့ **UID** ကို **1500** အဖြစ် ပြောင်းလဲမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** က အောက်ပါအတိုင်းပါ။ **usermod -u 1500 tom**

၃။ **usermod -C** >> user နဲ့ ပတ်သက်တဲ့ **info** လေးတွေ တွဲထည့်ချင်ရင် သုံးပါတယ်။

ဥပမာ **tom** ဆိုတဲ့ user နဲ့အတူ **i am single** ဆိုတဲ့ **info** လေး တွဲထည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါ။ **usermod -C "i am single" tom /etc/passwd** ဆိုတဲ့ file ထဲ ဝင်ကြည့်ရင် **GECOS** နေရာမှာ **i am single** ဆိုတာ ပေါ်နေပါလိမ့်မယ်။

၄။ **usermod -a -G >> user ကို တစ်ခုး group** ကနေ မဖယ်ရှားပဲ၊ **supplementary group** တစ်ခုခဲ့ **member** အဖြစ် သွင်းချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ကို **teamone** ဆိုတဲ့ **group** ထဲ သွင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **usermod -a -G teamone tom**

၅။ **usermod -g >> group** တစ်ခုကို **user** ရဲ့ **primary group** အဖြစ် သတ်မှတ်ချင်တဲ့ အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ရဲ့ **account** ဆောက်စဉ်က တွဲလျက်ပါလာတဲ့ **primary group** က **tom** ပါ။ **teamone** ဆိုတဲ့ **group** ကို **primary group** အဖြစ် ပြောင်းလဲ သတ်မှတ် မယ်ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **usermod -g teamone tom**

၆။ **usermod -d >> user ရဲ့ home directory** ကို ပြောင်းလဲ သတ်မှတ်ပေးချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ရဲ့ **account** ဖွင့်စဉ်က တွဲလျက်ပါလာတဲ့ **home directory** က **/home/tom** ပါ။ အဲဒီအစား **/home/www** ကို **tom** ရဲ့ **home directory** အဖြစ် ပြောင်းလဲ သတ်မှတ်ချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။

usermod -d /home/www tom

၇။ **usermod -m -d >> user ရဲ့ home directory** ကို တည်နေရာ ရွှေချင်တဲ့အခါ သုံးပါ တယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ရဲ့ **account** ဖွင့်စဉ်က တွဲလျက်ပါလာတဲ့ **tom** ဆိုတဲ့ **home directory** ရဲ့ တည်နေရာက၊ **/home** ရဲ့ အောက်မှာပါ။ ဒါကို **/home/www** ဆို **tom** ရဲ့ **home directory** ကို ရွှေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။

usermod -m -d /home/www tom

home directory ကို ပြောင်းလဲ သတ်မှတ်တာနဲ့ ရွှေ့တာဟာ မတူပါဘူး။ ကိုယ်တိုင် စမ်းသပ်ကြည့်ပါ။

၈။ **usermod -s >> user ရဲ့ default shell** ကို ပြောင်းလဲ သတ်မှတ်ချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ရဲ့ **default shell** က **bash shell** ပါ။ ဒါကို **sh shell** အဖြစ် ပြောင်းလဲ သတ်မှတ်ချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။

usermod -s /bin/sh tom

၉။ **usermod -L >> user account** တစ်ခုကို **lock** ချချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user account** ကို **lock** ချမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါ အတိုင်းပါ။ **usermod -L tom**

၁၀။ **usermod -U >> lock** ချခံထားရတဲ့ **user account** တစ်ခုကို **unlock** ပြန်လုပ်မယ့် အခါသုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user account** ကို **unlock** ပြန်လုပ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **usermod -U tom**

၁၁။ **gpasswd -d >> user** ကို **supplementary group** တစ်ခုထဲကနေ ပြန်ဖယ်ရှားချင်တဲ့အခါ ဖြူတွင်ချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ကို **teamone** ဆိုတဲ့ **group** ထဲက နေဖယ်ရှားမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** က အောက်ပါအတိုင်းပါ။

gpasswd -d tom teamone

၁၂။ **normal user account** တစ်ခုကို **sudo** ဆိုတဲ့ **command** နဲ့ **superuser** ဖြစ်တဲ့ **root** ကဲ့သို့၊ အသုံးပြုခွင့်ပေးနိုင်ဖို့အတွက် အဲဒီ **user** ကို **wheel** ဆိုတဲ့ group ထဲထည့်ပေးရပါတယ် ဥပမာ **tom** ဆိုတဲ့ **account** ကို **root** ကဲ့သို့ လုပ်ပိုင်ခွင့်ရအောင် သတ်မှတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **usermod -a -G tom wheel**

ဒါဆို **tom** ဆိုတဲ့ account ဘာ **root account** ကသာ ဖွင့်ကြည့်နိုင်တဲ့ file တွေကို **sudo** ဆိုတဲ့ command ခံပြီး ဖွင့်နိုင်ပါပြီ။ **sudo** နဲ့ ဆိုင်တဲ့ **configuration** တွေကို **/etc/sudoers** မှာ ဝင်ကြည့် နိုင်ပါတယ်။ က ! ခုဆို အတော်လေး စုံသွားပါပြီနော်။ user account တစ်ခုနဲ့ သက်ဆိုင်တဲ့ အကြောင်းတွေကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(19-5-2018)

```

root@localhost:~#
File Edit View Search Terminal Help
[root@localhost ~]# usermod -a -G teamone tom
[root@localhost ~]# groups tom
tom : tom teamone
[root@localhost ~]# id tom
uid=1001(tom) gid=1001(tom) groups=1001(tom),1002(teamone)
[root@localhost ~]# usermod -a -G wheel tom
[root@localhost ~]# groups tom
tom : tom wheel teamone
[root@localhost ~]# gpasswd -d tom teamone
Removing user tom from group teamone
[root@localhost ~]# groups tom
tom : tom wheel
[root@localhost ~]# #Have A Good Time All My Facebook Friends (Yin Thu)
[root@localhost ~]#

```

group နှင့် သက်ဆိုင်သော

linux မှာ group တည်ဆောက်တဲ့ အကြောင်း တင်ပေးပြီးပါပြီ။ ဒီတစ်ခါ တည်ဆောက်ထားတဲ့ group တွေကို လိုအပ်ချက်အရ ဘယ်လို ပြင်ဆင်သလဲဆိုတာ လေ့လာကြည့်ရအောင်။ group တစ်ခုနဲ့ သက်ဆိုင်တာတွေကို ပြင်ဆင်ချင်တဲ့အခါ **groupmod** ဆိုတဲ့ **command** ကို **option** တွေနဲ့ တွေ့သုံးရပါတယ်။ စလိုက်ကြရအောင် !

၁။ **groupmod -g >> group** ရဲ့ **GID** (group id) ကို ပြောင်းလဲ သတ်မှတ်ချင်တဲ့အခါ သုံးပါတယ်။ ဥပမာ **teamone** ဆိုတဲ့ **group** ရဲ့ **GID** ကို **1500** လို့ ပြောင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **groupmod -g 1500 teamone**

၂။ **group -n >> group** ရဲ့ **name** ကို ပြောင်းလဲလိုတဲ့အခါ သုံးပါတယ်။ ဥပမာ **teamone** ဆိုတဲ့ **group** ကို **google** အဖြစ် **name** ပြောင်းလဲမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် **command** ကတော့ အောက်ပါအတိုင်းပါ။ **groupmod -n google teamone**

ပြောရရင် **group** နဲ့ ပတ်သက်လို့ ဒီ **option** နှစ်ခုပဲ အသုံးများပါတယ်။

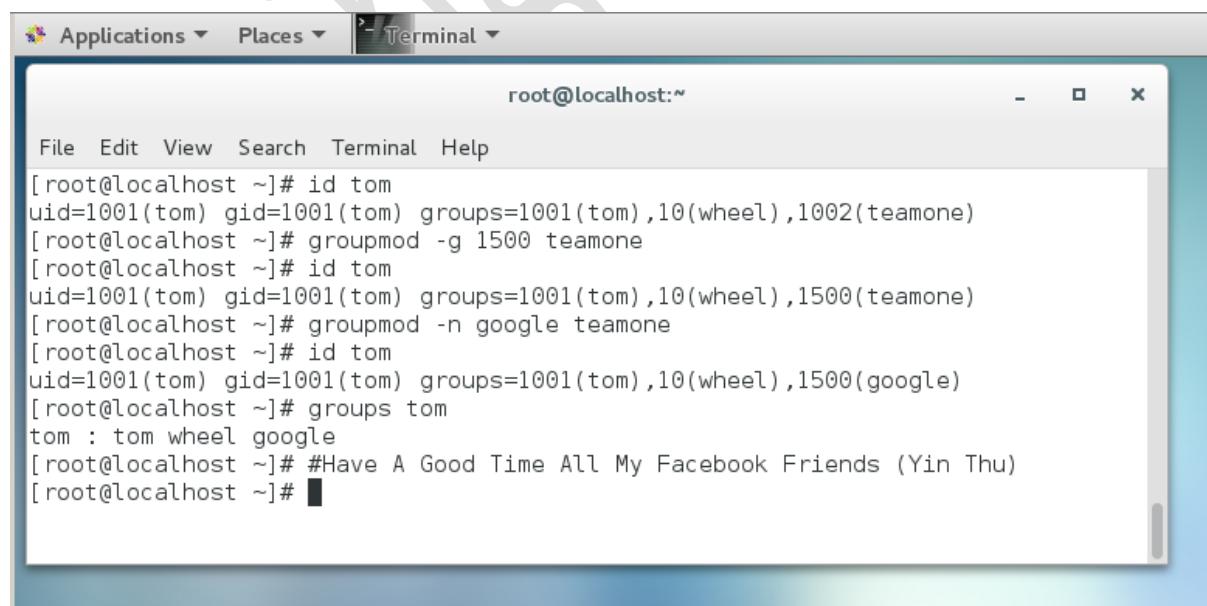
group နဲ့ သက်ဆိုင်တာတွေကို ပြင်ဆင်တဲ့ **groupmod** ဆိုတဲ့ **command** ရဲ့ အသုံးပြုပုံတွေကို **man groupmod** သို့မဟုတ် **groupmod --help** လို့ ရိုက်ပြီး ဆက်လက်လေ့လာနိုင်ပါတယ်။

group တွေကို **modify** လုပ် ပြင်ဆင်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(19-5-2018)



```

root@localhost ~]# id tom
uid=1001(tom) gid=1001(tom) groups=1001(tom),10(wheel),1002(teamone)
[root@localhost ~]# groupmod -g 1500 teamone
[root@localhost ~]# id tom
uid=1001(tom) gid=1001(tom) groups=1001(tom),10(wheel),1500(teamone)
[root@localhost ~]# groupmod -n google teamone
[root@localhost ~]# id tom
uid=1001(tom) gid=1001(tom) groups=1001(tom),10(wheel),1500(google)
[root@localhost ~]# groups tom
tom : tom wheel google
[root@localhost ~]# #Have A Good Time All My Facebook Friends (Yin Thu)
[root@localhost ~]#

```

superuser Vs normal user Vs Administrator

သိတေးသင့်တဲ့ အချက်လေးပေါ့။ super user, normal user နဲ့ administrator ရဲ့ကွဲပြားမှာ တွေကို လေ့လာကြည့်ရအောင်။

#superuser

၁။ linux system ပေါ့မှာ လုပ်ပိုင်ခွင့် အရှိခံး account ကို superuser လို့ ခေါ်ပါတယ်။ system file တွေကို configure လုပ်ခြင်း၊ user account များ ထပ်မံ တည်ဆောက်ခြင်း၊ ဖျက်ပစ်ခြင်း၊ password နှင့်ဆိုင်သော အချက်အလက်များကို manage လုပ်ခြင်း စတာတွေကို စီမံ ဆောင်ရွက် နိုင်ပါတယ်။ linux system မှာ root ဟာ တစ်ခုတည်းသော superuser account ပါပဲ။ သူအတွက် password ကို linux ကို စတင် install လုပ်စဉ်ကတည်းက သတ်မှတ်ပေးခဲ့ရပါတယ်။ [root@localhost~]# ဆိုတဲ့ **shell prompt** အဆုံးမှာ ရှိတဲ့ # သက်တာသည် ယခုအသုံးပြု နေသော **user** ဟာ **root account** ဖြစ်ကြောင်း ညွှန်းဆိုပါတယ်။

root account ကနေ **normal account** တစ်ခုလဲ ကူးပြောင်းချင်ရင် သို့မဟုတ် account တစ်ခုကနေတစ်ခုလဲ ကူးပြောင်းချင်ရင် **su** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ ဥပမာ **root account** ကနေ **tom** ဆိုတဲ့ **normal account** ဆီ ကူးပြောင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကအောက်ပါအတိုင်းပါ။ [root@localhost~]#**su tom**

ဒါက **superuser** ဖြစ်တဲ့ **root account** ရဲ့အကြောင်းပါ။

#normal_user

၂။ normal user ဆိုတာကတော့ အသစ် ထပ်မံတည်ဆောက်လိုက်တဲ့ user account တွေကို ဆိုလိုတာပါ။ root account လို့ လုပ်ပိုင်ခွင့်တွေ မရှိပါဘူး။ [tom@localhost~]\$ ဆိုတဲ့ **shell prompt** အဆုံးမှာ ရှိတဲ့ \$ သက်တာသည် ယခု အသုံးပြုနေသော **user** ဟာ **normal account** ဖြစ်ကြောင်း ညွှန်းဆိုပါတယ်။

normal account ကနေ **root account** ဆီ ကူးပြောင်းချင်ရင် **su -** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ account ကနေ **root** ဆိုတဲ့ **superuser account** ဆီ ကူးပြောင်းမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **su -**

#administrator

၃။ လိုအပ်ချက်အရာ **normal account** တစ်ခုကို **root account** ကဲသို့ ဆောင်ရွက်စေ လိုတဲ့အခါ **sudo** ဆိုတဲ့ **command** ကို အခြား **command** နဲ့ တွဲဖက် အသုံးပြုရပါတယ်။ ဒီလို အသုံးပြုနိုင်စွမ်းတဲ့ **account** ကို **administrator account** လို့ ခေါ်ပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **normal account** ဟာ **jerry** ဆိုတဲ့ **user account** ကို တည်ဆောက်မယ်ဆိုပါစို့။ **useradd jerry**

လို့ရှိက် **enter** ခေါက်လိုက်ရင် **Permission denied** ပြပါလိမ့်မယ်။ **Permission denied** မပြဖို့ **sudo** နဲ့ တွဲဖက်ပြီး၊ အသုံးပြုရမယ့် **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။

[tom@localhost ~]\$ sudo useradd jerry

သိထားရမှာက၊ **normal account** တစ်ခု အနေနဲ့ ခုလို **sudo** ဆိုတဲ့ **command** ကို အသုံးပြုပြီး **superuser** ကဲသို့ ပြုမှန်စွဲ၊ **root account** ကနေ အဲဒီ **normal user** ကို **wheel group** ထဲ သွင်းပေးရပါတယ်။ ဥပမာ၊ **root** အနေနဲ့ **tom** ဆိုတဲ့ **normal account** ကို **sudo** အသုံးပြုခွင့် ပေးမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

[tom@localhost ~]\$ usermod -a -G wheel tom

#sudo

၄။ **sudo** အသုံးပြုခွင့် ပေးခြင်းနဲ့ ဆိုင်တဲ့ အချက်အလက်တွေကို **/etc/sudoers** မှာ ဝင်ရောက် ဖတ်ရှုနိုင်ပါတယ်။ အသုံးပြုရမယ့် command ကတော့ **less /etc/sudoers** ပေါ့။

less /etc/sudoers

#edit_sudoers

၅။ တစ်ခု သတိပြုရမှာက၊ **sudo** အသုံးပြုခွင့်ပေးဖို့ အတွက် **normal account** တစ်ခုကို **wheel group** ထဲကို **add** ပြီးတာတောင်မှာ၊ **sudo** သုံးဖို့ အဆင်မပြောသေးရင်၊ **Permission denied** ပြနေသေးရင်၊ **/etc/sudoers** ထဲကို ဝင်ပြင်ပေးရပါမယ်။ ပြင်ပေးရမယ့် အဆင့်တွေကတော့ အောက်ပါအတိုင်းပါ။

- A. **vi /etc/sudoers** နဲ့ ဖွံ့ဖြိုးပါ။
 - B. **i** ကို နှိပ်ပြီး insert mode ထဲ ဝင်ပါ။
 - C. **#%wheel ALL=(ALL) ALL** ကို ရှာပါ။
 - D. **#%wheel ALL=(ALL) ALL** ရဲ့ ရှေ့ဆုံးက **#** သက်တကို ဖျက်ပစ်လိုက်ပါ။
 - E. **Esc** နှိပ်၊ **:wq!** နဲ့ ထွက်ပါ။ ဒါဆို အဆင်ပြောပါပြီ။ **normal account** တစ်ခုကို **wheel group** ထဲသွင်းပြီးတာတောင် **sudo** သုံးဖို့ အဆင်မပြောသေးရင် ဒီနည်းအတိုင်း ဖြေရှင်းနိုင်ပါတယ်။
- ကဲ ! စာလည်း နည်းနည်း ရှည်သွားပြီဗျာ။ super user, normal user နဲ့ administrator အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြောပါစေ။

Yin Thu

(20-5-2018)

```

File Edit View Search Terminal Help
[root@localhost ~]# su tom
[tom@localhost root]$ cd
[tom@localhost ~]$ useradd jerry
bash: /sbin/useradd: Permission denied
[tom@localhost ~]$ su -
Password:
Last login: Sun May 20 15:35:51 MMT 2018 on pts/0
[root@localhost ~]# usermod -aG wheel tom
[root@localhost ~]# id tom
uid=1001(tom) gid=1001(tom) groups=1001(tom),10(wheel),1500(google)
[root@localhost ~]# su tom
[tom@localhost root]$ cd
[tom@localhost ~]$ useradd jerry
bash: /sbin/useradd: Permission denied
[tom@localhost ~]$ sudo useradd jerry
[tom@localhost ~]$ sudo passwd jerry
Changing password for user jerry.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[tom@localhost ~]$ #Have A Good Time All My Facebook Friends (Yin Thu)
[tom@localhost ~]#

```

file နှင့် directory/folder များကို permission သတ်မှတ်ခြင်း

linux ပေါ်မှာ ဘာတွေ ရှိမလဲဆိုတော့ file တွေ directory တွေ ရှိပါမယ်။ အရေးကြီးတာတွေ ရှိသလို အရေး မကြီးတာတွေလည်း ရှိမယ်ပေါ့။ ဒီတော့ အရေးကြီးတာတွေကို မသက်ဆိုင်သူတွေ ရန်က ကာကွယ်ဖို့ လိုအပ်လာပြီပေါ့။ အဲဒီမှာ permission က အရေးပါလာပါပြီ။ linux system ကို ဝင်ရောက် အသုံးပြုနေတဲ့ user တွေ၊ တူညီတဲ့ group တစ်ခုရဲ့ member တွေ၊ group ထဲမှာ မပါတဲ့ တစ်ခြား other user တွေ အနေနဲ့ အဲဒီ file တွေ directory တွေကို access လုပ်ရာမှာ၊ သူတို့ရဲ့ အဆင့်အလိုက် permission တွေ ခွဲခြားသတ်မှတ် ပေးထားနိုင်ပါတယ်။ permission သတ်မှတ် နည်း method တွေဟာ ရှုပ်ထွေးသလို ထင်ရပေမယ့် သေချာလေ့လာကြည့်ရင် နားလည် ကျွမ်းကျင် သွားပါလိမ့်မယ်။

ကဲ ! စမ်းသပ်ဖို့ အတွက် **readme.txt** ဆိုတဲ့ **file** တစ်ခု ဆောက်လိုက်ပါ။

touch readme.txt ပြီးရင် **ls -l readme.txt** လို့ ရှိက် **enter** ခေါက်လိုက်တဲ့အခါ **-rw-r--r--** ဆိုတာတွေကို တွေ့ရပါမယ်။ အဲဒါက ဘာတွေလဲဆိုတော့ **permission** သတ်မှတ်ချက်တွေပါ။ permission တစ်ကြောင်းမှာ အပိုင်း ၃ ပိုင်းပါဝင်ပါတယ်။

၁။ file သို့မဟုတ် directory ကို ဖန်တီးတဲ့ **user** အတွက် သတ်မှတ်ထားတဲ့ **permission** >> **-rw-r--r-**
၂။ **group** ထဲမှာ ပါဝင်တဲ့ **user** တွေ အတွက် သတ်မှတ်ထားတဲ့ **permission** >> **-r-**

၃။ **file** ကို ဖန်တီးသူလည်း မဟုတ်၊ **group** ထဲမှာလည်း မပါတဲ့ **other user** တွေ အတွက် သတ်မှတ်ထားတဲ့ **permission** >> **-r-** စသဖြင့် ၃ ပိုင်း ခွဲခြားထားပါတယ်။

မှတ်ရလွယ်အောင် အတိုင်းပြောရရင်

- ၁။ **user (u)**
- ၂။ **group (g)**
- ၃။ **other (o)** ဆိုတဲ့ အပိုင်း ၃ ပိုင်းပေါ့။

ဥပမာ **tom** နဲ့ **jerry** ဆိုတဲ့ **user** နှစ်ဦးဟာ **hackergroup** ဆိုတဲ့ **group** ထဲမှာ **member** အဖြစ်ပါတယ်။

user >> အဲဒီ **tom** ဆိုတဲ့ **user** က **readme.txt** ဆိုတာလေးကို တည်ဆောက်လိုက်မယ် ဆိုပါစွာ။
tom ဆိုတဲ့ **user** ဟာ အဲဒီ **file** လေးရဲ့ **owner** အဖြစ် **readme.txt** ကို **access** လုပ်နိုင်မယ်။

group >> **jerry** သည်လည်း **hackergroup** ဆိုတဲ့ **group** ထဲမှာ ပါတဲ့ အတွက် **member** တစ်ဦး အဖြစ် **readme.txt** ကို **access** လုပ်နိုင်မယ်။

other >> **owner** လည်း မဟုတ် **group** ထဲမှာလည်း မပါတဲ့ **other user** တွေသည်လည်း သူတို့ အတွက် သတ်မှတ်ပေးထားတဲ့ **permission** အရ **readme.txt** ကို **access** လုပ်နိုင်မယ်။ ဒီသဘောပါ။

ဥပမာ၊ အဲဒီ **readme.txt** ဆိုတာလေးကို **ls -l readme.txt** ဆိုပြီး ခေါ်ကြည့်လိုက်ရင် **rw-r--r--** ဆိုပြီး တွေ့ရပါမယ်။

ရှေ့ခံးက **rw-** က ဖန်တီးတဲ့ **user** (**tom**) အတွက် သတ်မှတ်ထားတဲ့ **permission** ပါ။ ဒုတိယ **r--** က **group** ထဲမှာပါတဲ့ **user** တွေ (**jerry**) အတွက် သတ်မှတ်ထားတဲ့ **permission** ပါ။ တတိယ **r--** ကတော့ **owner** လည်း မဟုတ် **group** ထဲမှာလည်း မပါတဲ့၊ **other user** တွေ အတွက် သတ်မှတ်ထားတဲ့ **permission** ပါ။ ဒါကို permission သတ်မှတ်ချက် လို့ ခေါ်ပါတယ်။

permission ၃ မျိုးရှိပါတယ်။ အဲဒီတွေကတော့၊

၁။ **read (r)** >> numeric တန်ဖိုးအားဖြင့် 4

၂။ **write (w)** >> numeric တန်ဖိုးအားဖြင့် 2

၃။ **execute (x)** >> numeric တန်ဖိုးအားဖြင့် 1 စသဖြင့် ရှိပါတယ်။

အတို့ ပြန်ချုပ်ရရင် **permission** တစ်ကြောင်းမှာ **user, group, other** ဆိုတဲ့ အပိုင်း ၃ ပိုင်း ရှိပြီး၊ **permission** က **read, write, execute** ဆိုပြီး ၃ မျိုး ရှိတာပါ။ **permission** ပေးတဲ့အခါ **chmod** ဆိုတဲ့ command ကို အသုံးပြုပါတယ်။ **chmod**

permission ပေးနည်းကတော့

၁။ **symbolic** နဲ့

၂။ **numeric** ဆိုပြီး ၂ နည်း ရှိပါတယ်။

#symbolic_method

၁။ **symbolic method** :: symbolic ဖြင့် permission ပေးတယ် ဆိုတာကတော့ **read (r), write (w), execute (x)** စတဲ့ သတ်မှတ်ချက်တွေကို **rwx** စသဖြင့် အတို့ကောက် သတ်မှတ်ပြီး ပေးတဲ့ နည်းပါ။

file တစ်ခုကို **user** အတွက်သာ သို့မဟုတ် **group** အတွက်သာ သို့မဟုတ် **other** အတွက်သာ၊ **permission** တိတိကျကျ သတ်မှတ်ပေးချင်တဲ့ အခါ = သက်တကို သုံးပါတယ်။

ဥပမာ **readme.txt** ကို **user** အတွက် **read** **write** **execute** ဆိုတဲ့ **permission**, **group** အတွက် **read** **write** ဆိုတဲ့ **permission**, **other** အတွက် **read only** ဆိုတဲ့ **permission**၊ စသဖို့
တစ်ခုချင်း ကွက်ပြီး တိတိကျကျ သတ်မှတ်ပေးမယ် ဆိုပါစ္စား = နဲ့ တွဲပြီး အောက်ပါအတိုင်း
တစ်ဆင့်ချင်း သတ်မှတ်သွားရမှာပါ။

user (u) အတွက် >> **chmod u=rwx readme.txt**
group (g) အတွက် >> **chmod g=rw readme.txt**
other (o) အတွက် >> **chmod o=r readme.txt**
အားလုံး (a) အတွက် >> **chmod a+x readme.txt**

readme.txt ကို **ls -l readme.txt** ဆိုပြီး ပြန်ခေါ်ကြည့်လိုက်တဲ့ အခါ **rwxr--r--** လို့ တွေ့ရပါမယ်။
- သက်တကတော့ permission တစ်ခုမှာ **rwx** ဆိုပြီး ရှိရမှာဖြစ်ပြီး၊ ကွက်လပ်ဖြစ်နေတဲ့
နေရာတွေကို - နဲ့ ဖြည့်ထားတဲ့ သဘောပါ။ ပေးထားတဲ့ permission ထဲက ပြန်လည့်ချင်တဲ့ အခါ -
သက်တကို သုံးပါတယ်။ ဥပမာ စောစောက **file1** မှာ **user** အတွက် **read** **write** **execute** လို့
ပေးထားခဲ့တယ်။ အဲဒီထဲကမှ **execute** ကို ပြန်ဖြုတ်ချင်တယ် ဆိုပါစ္စား။ - နဲ့ တွဲပြီး အောက်ပါအတိုင်း
သတ်မှတ်သွားရမှာပါ။ **user (u)** အတွက် >> **chmod u-x readme.txt**

ဥပမာ စောစောက **readme.txt** မှာ **group** အတွက် **read** **write** လို့ ပေးထားခဲ့တယ်။ အဲဒီထဲကမှ
write ကို ပြန်ဖြုတ်ချင်တယ် ဆိုပါစ္စား။ - နဲ့ တွဲပြီး အောက်ပါအတိုင်း သတ်မှတ်သွားရမှာပါ။
group (g) အတွက် >> **chmod g-w readme.txt**

ဥပမာ စောစောက **readme.txt** မှာ **other** အတွက် **read** လို့ ပေးထားခဲ့တယ်။ အဲဒီထဲကမှ **read**
ကို ပြန်ဖြုတ်ချင်တယ်၊ ဘာ permission မှ မပေးတော့ဘူး။ **access** လုပ်ခွင့် မပေးတော့ဘူး ဆိုပါစ္စား။
- နဲ့ တွဲပြီး အောက်ပါအတိုင်း သတ်မှတ်သွားရမှာပါ။

other (o) အတွက် >> **chmod o-r readme.txt**

ဟုတ်ပြီး **readme.txt** ကို **ls -l readme.txt** ဆိုပြီး ပြန်ခေါ်ကြည့်လိုက်တဲ့ အခါ **rw-----** လို့
တွေ့ရပါမယ်။

ပေးထားတဲ့ permission ထဲကို ထပ်ပေါင်းပေးချင်တဲ့အခါ + သက်တကို သုံးပါတယ်။

ဥပမာ **readme.txt** မှာ **user (u)** အတွက် **read** **write** လို့ ပေးထားတာကို **execute (x)** ဆိုတာလေး ထပ်ပေါင်းထည့်ပေးချင်တယ် ဆိုပါစို့။ + နဲ့ တွဲပြီး အောက်ပါအတိုင်း သတ်မှတ်သွားရမှာပါ။

user (u) အတွက် >> **chmod u+x readme.txt**

မှတ်စရာ အနေနဲ့ကာ ၁။ **user (u)** ၂။ **group (g)** ၃။ **other (o)** ဆိုတဲ့ အပိုင်း ၃ ပိုင်းလုံးကို တစ်ပြိုင်နှင်း ထပ်ပေါင်းထည့်ပေးချင်တယ် ဆိုပါစို့။ အောက်ပါအတိုင်း သတ်မှတ်သွားရမှာပါ။

ဥပမာ **readme.txt** ဆိုတဲ့ file ကို **user (u)** အတွက် **read**, **write** and **execute (rwx)** ၁ **group** အတွက် **rwx** ၁ **other (o)** အတွက်လည်း **rwx** သတ်မှတ်မယ် ဆိုပါစို့။ **a** နဲ့ တွဲပြီး အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါတယ်။ အားလုံး အတွက် >> **chmod a=rwx readme.txt**

ဥပမာ **readme.txt** ဆိုတဲ့ file ကို **user (u)** အတွက် **execute** **group** အတွက် **execute** **other** **(o)** အတွက်လည်း **execute** ထပ်ပေါင်း သတ်မှတ်မယ် ဆိုပါစို့။ **a** နဲ့ တွဲပြီး အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါတယ်။ **chmod a+x readme.txt** သို့မဟုတ် **chmod +x readme.txt**

ကဲ ! ခုလောက်ဆို symbolic နည်းကို သုံးပြီး permission ပေးတာကို သဘောပေါက်လိမ့်မယ် ထင်ပါတယ်။ ၂ ခေါက်လောက် ထပ်ဖတ်ကြည့်ဖို့ အဆိုပြုပါရစေ။

#numeric_method

၂။ **numeric method** :: numeric ဖြင့် permission ပေးတယ်ဆိုတာကတော့ စောစောက အသုံးပြုတဲ့ **rwx** နေရာမှာ numeric တန်ဖိုး တွေ အစားထိုးပြီး သတ်မှတ်တာကို ဆိုလိုတာပါ။ အစပြန်ကောက်ရရင် permission ၃ မျိုးရှိပါတယ်။ အဲဒါတွေကတော့၊

read (r) numeric တန်ဖိုးအားဖြင့် 4

write (w) numeric တန်ဖိုးအားဖြင့် 2

execute (x) numeric တန်ဖိုးအားဖြင့် 1 စသဖြင့် ရှိပါတယ်။

အဲဒါ 4 (r) ... 2 (w) ... 1 (x) ဆိုတဲ့ numeric တန်ဖိုးတွေကို ပေါင်းစပ်ပြီး permission သတ်မှတ်တာကို numeric method လို့ ခေါ်တာပါ။

ဥပမာ A မှ **readme.txt** မှာ **user** အတွက် full control သတ်မှတ်ချက် ကို symbolic method နဲ့
ပေးမယ်ဆို **rwx** လို့ သတ်မှတ်ပေးရမှာပါ။ အဲဒီအစား numeric method ကို သုံးမယ်ဆို **4 (r) + 2
(w) + 1 (x) = 7** ရတာကြောင့် **rwx** အစား **7** လို့ ရေးရပါတယ်။

ဥပမာ B မှ **readme.txt** မှာ **group** အတွက် **read** **write** သတ်မှတ်ချက် ကို symbolic method နဲ့
ပေးမယ်ဆို **rw** လို့ သတ်မှတ်ပေးရမှာပါ။ အဲဒီအစား numeric method ကို သုံးမယ်ဆို **4 (r) + 2
(w) = 6** ရတာကြောင့် **rw** အစား **6** လို့ ရေးရပါတယ်။

ဥပမာ C မှ **readme.txt** မှာ **other** အတွက် **read** only သတ်မှတ်ချက် ကို symbolic method နဲ့
ပေးမယ်ဆို **r** လို့ သတ်မှတ်ပေးရမှာပါ။ အဲဒီအစား numeric method ကို သုံးမယ်ဆို **4 (r) = 4**
ရတာကြောင့် **r** အစား **4** လို့ ရေးရပါတယ်။

ဥပမာ ABC သုံးချပောင်းပြီး **readme.txt** ကို numeric method နဲ့ permission သတ်မှတ်ပေး
ရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

user, group, other သုံးခဲ့လုံး အတွက် >> **chmod 764 readme.txt**
ဆက်လွှဲလာ ကြည့်ရအောင်။

numeric method အရာ **4 (r) ... 2 (w) ... 1 (x)** စတဲ့ တန်ဖိုးတွေကို ပေါင်းစပ်လိုက်တဲ့ အခါ
အောက်ပါအတိုင်း permission (စ) မျိုး ရရှိပါတယ်။

7 : full control, **read, write, execute** >> **rwx = 4+2+1 = 7**

6 : **read and write** >> **rw = 4+2 = 6**

5 : **read and execute** >> **rx = 4+1 = 5**

4 : **read only** >> **r = 4 = 4**

3 : **write and execute** >> **wx = 2+1 = 3**

2 : **write only** >> **w = 2 = 2**

1 : **execute only** >> **x = 1 = 1**

0 : **none** >> **- - - = 0**

```

File Edit View Search Terminal Help
[root@localhost Desktop]# touch readme.txt
[root@localhost Desktop]# ls -l readme.txt
-rw-r--r--. 1 root root 0 May 20 15:40 readme.txt
[root@localhost Desktop]# chmod 750 readme.txt
[root@localhost Desktop]# ls -l readme.txt
-rwxr-x--. 1 root root 0 May 20 15:40 readme.txt
[root@localhost Desktop]# chmod 751 readme.txt
[root@localhost Desktop]# ls
readme.txt
[root@localhost Desktop]# ls -l readme.txt
-rwxr-x--. 1 root root 0 May 20 15:40 readme.txt
[root@localhost Desktop]# #Have A Nice Day All My Facebook Friends (Yin Thu)
[root@localhost Desktop]#

```

symbolic VS numeric

ဥပမာ numeric method ကို အသုံးပြုပြီး **readme.txt** ကို **user** အတွက် **read and write (rw)**,
group အတွက် **read and execute (rx)**, **other** အတွက် **read only (r)** ပေးမယ် ဆိုပါစွဲ။
အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **chmod 654 readme.txt**

ဥပမာ symbolic method ကို အသုံးပြုပြီး **readme.txt** ကို user အတွက် read and write (rw), group အတွက် read and execute (rx), other အတွက် read only (r) ပေးမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတေသာ အောက်ပါအတိုင်းပါ။

user (u) အတွက်	>> chmod u=rw readme.txt
group (g) အတွက်	>> chmod g=rx readme.txt
other (o) အတွက်	>> chmod o=r readme.txt

permission သတ်မှတ်ပေးပြီးရင် **ls -l readme.txt** လို့ ရိုက် enter ခေါက်ပြီး စစ်ဆေးကြည့် လိုက်ပါဉိုး။ ဒီနေရာမှာ method နှစ်ခုရဲ့ကွားချက်အနေနဲ့ကာ symbolic method သုံးရင် user, group, other စတာတွေကို တစ်ကြောင်းချင်း ခွဲပေးရတာဖြစ်ပြီး numeric method သုံးရင်တော့ တစ်ကြောင်းတည်းနဲ့ ကိစ္စ ပြီးမြောက်နိုင်တာပါပဲ။

permission for directory

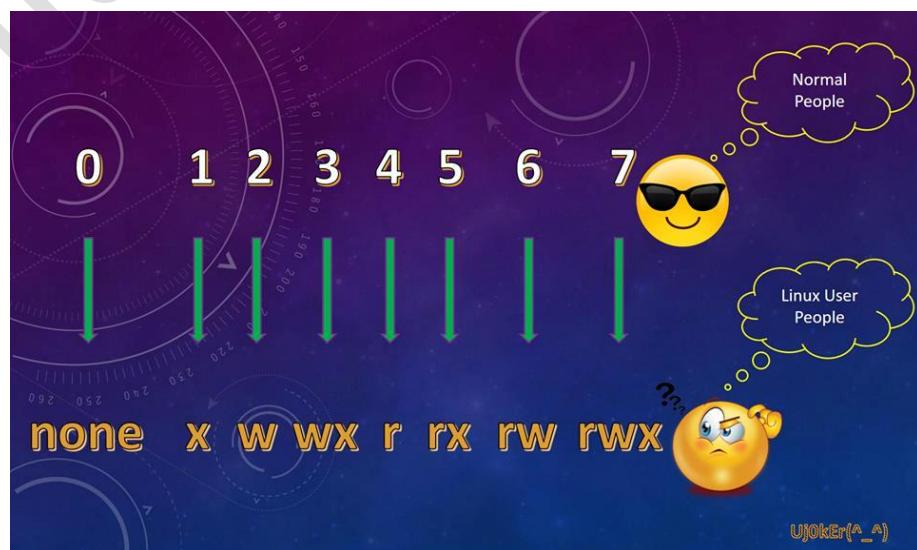
file တွေနဲ့ အလားတူပါပဲ၊ directory တွေကိုလည်း symbolic method နဲ့ numeric method သုံးပြီး permission သတ်မှတ်ပေးနိုင်ပါတယ်။

permission ပေးပြီးရင် directory တွေကို ပြန်စစ်ရမယ့် command ကတေသာ **ls -ld dirname** ပါ။ နောက်တစ်ချက်က၊ directory တစ်ခုကို permission ပေးတဲ့အခါ သူအောက်မှာ ရှိတဲ့ file တွေ sub directory တွေကိုပါ သက်ရောက်သွားစေချင်ရင် **chmod** ကို **-R** ဆိုတဲ့ option နဲ့ တွဲသုံးနိုင်ပါတယ်။ **chmod -R 750 directoryname** စသဖြင့်ပေါ့။

file တွေ directory တွေကို permission သတ်မှတ်ပေးပဲပေးနည်းတွေကတော့ ဒီလောက်ပါပဲ။ အနည်းငယ် ရှုပ်ထွေးသလို ရှိတာကြောင့် တစ်ကြိမ်တည်းနဲ့ မရှင်းလင်းရင်၊ အစအဆုံး ၃ ကြိမ်လောက် ပြန်ဖတ်ပါ။ စမ်းသပ်ကြည့်ပါ လို့ အဆိုပြုပါရစေ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(25-5-2018)



Red Hat Linux Exam တွေဖြစ်အခါ အရင်ဆုံးစဖြေရတာ၊ RHCSA (Red Hat Certified System Administrator) ဆိုတဲ့ Exam ပါ ..

ဖြေဆိုဖို့အတွက် OS level administration ပိုင်း တော်တော်များများကို နားလည်ထားပြီး ကျမ်းကျင့်ဖို့လိုပါတယ် ..

File System .. File path တွေကစလို့ File Editing အကြောင်းတွေ User and Group အကြောင်းတွေ Permission အကြောင်းတွေကို ကျမ်းကျင့်ပိုင်နိုင်ရပါတယ် ..

(Permission ဆိုတဲ့နေရာမှာလည်း Read Write Execute လောက်ပေးတတ်ရုံး, chmod 777 လေးပေးတတ်ရုံးမရပါဘူး။)

Access Control Lists (ACLs) အကြောင်းတွေပါ ကျမ်းကျင့်နားလည်ထားရပါတယ် ..

Service Daemon အကြောင်းတွေလည်းသိထားဖို့လိုပါတယ်။ Service တခု start stop လုပ်ဖို့တွေ services တွေကို control လုပ်နိုင်ဖို့တွေလိုပါတယ်။

နောက်ပိုင်း Logs အကြောင်းတွေ System Process အကြောင်းတွေ software Packages manage အကြောင်းတွေပါ ထဲထဲဝင်ဝင်သိဖို့လိုပါတယ် ..

(yum install <package> လေး run တတ်ရုံးမရပါဘူး .. local repo ဆောက်တတ်ဖို့ပါလိုအပ်ပါတယ်။)

ဒါတော် အခြေခံအပိုင်းပုံးပါသေးတယ် ..

နောက်ထပ် Future Schedule Tasks run တာတွေ SELinux အကြောင်းတွေ LDAP server အကြောင်းတွေ သိဖို့လိုပါတယ်။

Partitions ဆိုရင်လည်း ထဲထဲဝင်ဝင်သိဖို့လိုပါတယ်။ Command Line ကနေ partitions ပိုင်းတာတွေ LVM ပိုင်းတာတွေ Swap partition ပိုင်းတာတွေ File System Mounting အပိုင်းတွေပါ သိဖို့လိုပါတယ်။

Mounting အပိုင်းမှာဆိုရင်လည်း mount command တခုထဲ သုံးတတ်ရုံးမရသေးပါဘူး ..

Temporary mount , Permanent mount, auto mount ပိုင်းတွေကအစ ထဲထဲဝင်ဝင်သိဖို့ကိုလိုတာပါ ..

Network File System (NFS) အကြောင်းတွေ Server Message Block (SMB) အကြောင်းတွေကိုပါ ထဲထဲဝင် ကျမ်းကျင့်နေဖို့လိုပါတယ်။ Boot Troubleshooting ပိုင်းဆိုလည်း အတော်ကျမ်းကျင့်ဖို့လိုပါတယ်။ တခါတလေ Exam မှာ Boot error တတ်တဲ့အခါ မဖြေရှင်းနိုင်လို့ ကျသွားတဲ့ လူတွေလည်းမနည်းပါဘူး။ Boot Troubleshooting က အပြင် Real World မှာဆိုလည်း အတော်ကိုလိုအပ်တဲ့ ကဏ္ဍတာပါ။ (ဒါတော် စာရှည်မှာဆိုတာကြောင့် တချို့ဟာတွေ ထည့်မရေးထားပါဘူး ..

အခိုက ဆိုလိုချင်တဲ့ ရည်ရွယ်ချက်က RHCSA exam ဖြစ်ဖို့ရင် OS level Administration ပိုင်းကို နှုန်းစပ်စပ် ကျမ်းကျင့်ဖို့ နားလည်ဖို့ practice ရှိဖို့တွေ အများကြီးလိုပါတယ်။
ဒါမှာလည်း certified ဖြစ်သွားတာနဲ့အညီ skill ပါရသွားမှာပါ ။။။)

RHCSA exam ဖြစ်နိုင်ဖို့ရင် Web server configure ချတတ်ဖို့ DHCP DNS Mail .. etc ..server ပိုင်း install , configure ချတတ်ဖို့ဆိုတာထက် OS level administration ပိုင်းကျမ်းကျင့်မှ ဖြေလို့ရတာပါ ..

(ဒါတောင် တချို့ ကျောင်းသားတွေက Linux ပေါ်မှာ Web,DHCP,DNS,Mail server တွေသာ install configure ချတတ်တာပဲရှိတာပါ .. ဘယ် command ကာဘာကိုဆိုလိုတယ် file path တွေက ဘယ်လို့ဉာဏ်နဲ့ရတယ် .. configuration file က ဘယ်နားမှာဆိုတာတွေတောင် မသိပါဘူး .. သင်တန်းတချို့ရဲ့ကောင်းမှုကြောင့် Linux administration အခြေခံမရှိပဲ ဒီ cmd ရှိက်, ဒါ run, ပြီးရင် server up ပါးဆိုပြီး သင်လိုက်တော့ တိုင်ပတ်ကုန်ရော ..)

တချို့နေရာတွေမှာ ကြော်လားတွေကြည့်ရတာ အဆင်မပြုလို့ ပြောပြတာပါ ..

Web Server, DNS, DHCP တွေသင်ပြီး အောက်မှာရေးထားတော့ RHCSA ဖြစ်လို့ ရပါတယ်ဆိုပြီး ရေးထားတာတွေ တွေ့ရလိုပါ ..

(ဖြစ်ဖို့ အထောက်အကူးပြုတယ်လို့ ရေးရင်တောင် အဆင်ပြေပါသေးတယ်)

Red Hat exam အကြောင်းမသိရင်လည်း Google မှာရှာပြီး ဖတ်စေချင်ပါတယ်... .

မသိတဲ့ကျောင်းသားတွေဆိုရင် နားလည်မှုလွှဲပြီး အမှတ်မှားမှာဆိုလိုပါ .. ကျန်တဲ့ ဘာစိတ်မှတော့ မရှိပါဘူး .. ။ (Certificate name တောင် မှန်အောင် မပေါင်းထားပဲ ကြော်လားတာတွေလည်း အများကြီးပါ .. xD) ထားပါ .. နောက်တခုက Linux ကိုလေ့လာလိုတဲ့ သူတွေဆိုရင်လည်း Linux ရဲ့ system administration skill အရင်လေ့လာဖို့ suggest လုပ်ချင်တာပါ ..

linux ကို install တောင် ဖြောင့်အောင်မလုပ်နိုင်သေးပဲ .. Linux command တွေထဲက ဘယ် cmd ကဘာလဲ ဘာအဓိပါယ်လဲ .. Linux File System ကဘယ်လိုရှိတာလဲ .. ဆိုတာတွေတောင် မသိပဲ Linux ပေါ်မှာ Server install configure သွားလုပ်ရင် သေချာပြောရဲပါတယ် .. ဘာမှ နားမလည်ပဲ စိတ်ပျက် လက်ပျက်ဖြစ်သွားမှာပါ .. ပြောချင်တာကတော့ ဒါပါပဲ .. တချို့လွှဲနေတာလေးတွေရယ် Linux ကိုလေ့လာချင်ပေမယ့် လမ်းမှားနေတဲ့သူလေးတွေရယ် တွေ့တွေ့နေရလို့ တတ်သလောက် မှတ်သလောက် ထောက်ပြတဲ့အနေနဲ့ ရေးလိုက်တာပါ .. မကျန်ပဲလို့ ဆွေးနွေးချင်လည်း ရပါတယ် ac လေးကိုသာ report မထုပါနဲ့ .. xD credit - #YeLinAung



Linux Shell Scripting

တခိုက် linux ရဲ shell scripting အကြောင်း တင်ပေးဖို့ တောင်းဆိုကြတယ်။ အားတော့ tuto တင်တာ စော နားပြီး shell scripting အကြောင်း အလာပသလာပ ပြောကြည့်ရအောင်နော်။ shell scripting ဆိုတာ ဘာလဲဆိုတော့ကာ၊ တခါတလေမှာ မိမိ လိုချင်တဲ့ result တစ်ခု ရဖို့ အတွက် command တစ်ကြောင်းတည်းနဲ့ အဆင်မပြေား။

အဲဒီအခါ မိမိ သုံးစွဲမယ့် command တွေကို file တစ်ခုတည်းမှာ ရှိက်ထည့်ပြီး အဲဒီ file ကို terminal ကနေတဆင့် shell ပေါ်မှာ run ခိုင်းရပါတယ်။

အဲဒီလို run နိုင်ဖို့ command တွေကို အစီစဉ်တကျ ရေးသွင်းတာကို shell scripting လို ခေါ်ပါတယ်။

script ရေးရာမှာ work done ဖြစ်ဖို့ အတွက် programming concept တွေကိုလည်း ဟန်ချက်ညီညီ ထည့်သွင်း အသုံးပြုရပါတယ်။ ဥပမာ ->

```
#!/usr/bin/bash
```

```
echo "Enter your user name: $1 "
```

```
if [ "$1" = "mg mg" ]; then
```

```
printf "hello $1"
```

```
else
```

```
printf "The user name you entered is incorrect"
```

ပြီးရင် file name နောက်မှာ .sh နဲ့ အဆုံးသတ်ပြီး save ရပါတယ်။ **myfirstshell.sh** ဆိုပါစ္စ။ run ချင်ရင်တော့ ./myfirstshell.sh ဆိုပြီး run ရပါတယ်။ ./myfirstshell.sh

code တွေ ဖြစ်ပေမယ့် သူ့ကို program လို မခေါ်ပဲ ဘာလို့ shell script လို ခေါ်ဘာလဲဆိုတော့၊ သူက ဘာလုပ်ရင် ဘာဖြစ်မယ်ဆိုတဲ့ အသင့် ရှိပြီးသား command တွေကို ယူသုံးတာရယ်၊ instruction code အတိုလေးတွေ ဖြစ်တာရယ်၊ shell ကို အားပြုရတာရယ်ကြောင့် shell script လို ခေါ်ရတာပါ။

shell script ရေးနိုင်ဖို့ အတွက် ဘာတွေ သိတေးရမလဲ ဆိုတော့ -->

၁။ **linux system administration** ကို ဦးစွာ ကျမ်းကျင်ရပါမယ်။

၂။ **programming concept** ရှိရပါမယ်။ **variable** အကြောင်း, **statement** တွေ အကြောင်း,

array တွေအကြောင်း စသည်ဖြင့်ပေါ့။

ဘယ်လို လေ့လာမလဲ?

၁။ linux system administration ကို လေ့လာချင်ရင်တွေ Linux Ninja - Myanmar က တင်ပေးနေတဲ့ post တွေကတဆင့် google ရဲ့ အကူအညီလေးနဲ့ တွဲဖက် လေ့လာနိုင်ပါတယ်။ ဒါမှမဟုတ် သင်တန်းကျောင်းတွေ တက်ပေါ့။

၂။ programming concept မရှိခဲ့ရင် javascript လို့ language တစ်ခုကို လေ့လာမယ်ဆို concept ကောင်းတွေ ရရှိမှာပါ။ ဒါမှမဟုတ် linux shell scripting စာအုပ်တွေကနေလည်း လေ့လာ နိုင်ပါတယ်။

ဟုတ်ပြီ shell script တွေ ရေးတတ်ဖို့ အရေးကြီးလား?

၁။ ဆိုတော့ကာ အရေးမကြီးပါဘူး။ linux ကို normal user အဖြစ် အသုံးပြုမယ်ဆိုရင်ပေါ့။

၂။ professional အဖြစ် အသုံးပြုမယ်ဆိုရင်တော့ အရေးကြီးပါတယ်။

shell script ကို ဘာအတွက် သုံးလဲ?

၁။ များပြားလှတဲ့ config တွေကို server တွေမှာ ထည့်သွင်းရမယ့်အခါ အချိန်ကုန် သက်သာစေဖို့ အတွက်။

၂။ auto backup အတွက်။

၃။ plugin တွေ ရေးဖို့ အတွက်။

စသည်ဖြင့်ပေါ့ shell script ကို အသုံးပြုပါတယ်။

ကဲ ! ခုလောက်ဆို linux ရဲ့ shell scripting အကြောင်း concept တခါး။ ရသွားမယ် ထင်ပါတယ်။

ပိုမို သိချင်ရင်တော့ google ပေါ့။

အလပ်သလာပ ပြောတာလေး ဒီတင်နားပါရစေ။

အားလုံး အဆင်ပြေပါစေ။

PS :: စာဖတ်သူတို့ရဲ့ knowledge လေးတွေ၊ အမြင်လေးတွေ၊ အယူအဆလေးတွေလည်း comment မှာ ချိန်ရှစ်ပေးခဲ့ကြပါဦး။ အပန်းမကြီးရင်ပေါ့။

Yin Thu

(26-5-2018)

Linux™
SHELL
SCRIPTING



special permission

file နဲ့ folder တွေကို permission ပေးတဲ့အကြောင်း ပြီးသွားတော့၊ ခုတစ်ခါ special permission တွေ အကြောင်း ဆက်လိုက်ရအောင်။
ခက်ခက်ခဲ့တော့ မဟုတ်ပါဘူး။

chmod ဆိုတဲ့ command နဲ့ တွဲလျက် -->

၁။ user (u) အတွက် special permission (s) : u+s

၂။ group (g) အတွက် special permission (s) : g+s

၃။ other (o) အတွက် sticky (t) : o+t

ဆိုပြီး ၃ မျိုး ရှိပါတယ်။

၁။ u+s >> အဲဒီ file ကို ပိုင်ဆိုင်တဲ့ user အဖြစ် run ပါတယ်။

ဥပမာ **tom** ဆိုတဲ့ user က **readme.txt** ဆိုတဲ့ file ကို ထောက်လိုက်တယ်။ ပြီးရင် **chmod u+s**

readme.txt လို့ user နဲ့ ဆိုင်တဲ့ special permission ပေးလိုက်တယ် ဆိုပါစို့။

chmod u+s readme.txt

jerry ဆိုတဲ့ user က အဲဒီ **readme.txt** ကို လာ run မယ် ဆိုပါစို့။ အဲဒီ file ဟာ **tom** ဆိုတဲ့ user အနေနဲ့ run ပါတယ်။ ဒီသဘောပါ။

၂။ g+s >> group က ပိုင်ဆိုင်တဲ့ file အဖြစ် run ပါတယ်။

ဥပမာ **tom** ဆိုတဲ့ user က **dir1** ဆိုတဲ့ directory ကို ထောက်လိုက်တယ်။ အဲဒီ directory ကို **google** ဆိုတဲ့ group က ပိုင်တယ်ပေါ့။ ပြီးရင် **tom** က **chmod g+s dir1** လို့ group နဲ့ ဆိုင်တဲ့ special permission ပေးလိုက်တယ် ဆိုပါစို့။ **chmod g+s dir1**

jerry ဆိုတဲ့ user က အဲဒီ **dir1** ထဲမှာ file တစ်ခု လာထောက်မယ် ဆိုပါစို့။ အဲဒီ file ဟာ **google** ဆိုတဲ့ group က ပိုင်တဲ့ file အနေနဲ့ run ပါတယ်။ ဒီသဘောပါ။

၃။ o+t >> ထောက်လိုက်တဲ့ file တစ်ခုကို **chmod o+t** လို့ special permission သတ်မှတ် လိုက်တယ်ဆိုရင်၊ root နဲ့ owner ကလွှဲပြီး တစ်ခြား user တွေ အနေနဲ့ full control (rwx)

ရထားရင်တောင် အဲဒီ file ကို ဖျက်လို့ မရနိုင်ပါဘူး။ ဒီသဘောပါ။ **chmod o+t readme.txt**

တစ်ခု သိထားရမှာက ပေးလိုက်တဲ့ special permission s/g/t တစ်ခုခုဟာ၊ သာမဏ် permission

(rwx) ကာ x ရဲ့ နေရာကို ဝင်ယူတယ်ဆိုတာပါပဲ။ **နှစ်မှုလ permission** မှာ execute (x) ပေးပြီးသား

ဆိုရင် **small-letter** အသေးနဲ့ဖော်ပြုမှဖြစ်ပြီး၊ နိဂုံးလ permission မှာ execute (x) မပေးထားဘူးဆိုရင် **capital-letter** အကြီးနဲ့ဖော်ပြုပါလိမ့်မယ်။

ဥပမာ အနေနဲ့ ကြည့်ချင်ရင်တော့၊ **file** သို့မဟုတ် **folder** တစ်ခုကို special permission သတ်မှတ်ပြီးတိုင်း **file** အတွက် **ls -l** **folder** အတွက် **ls -ld** နဲ့ ခေါ်ကြည့်မယ်ဆိုရင် ပြောင်းလဲမှု တွေကို တွေ့မြင်ရပါလိမ့်မယ်။

special permission နဲ့ သက်ဆိုင်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(27-5-2018)

```
File Edit View Search Terminal Help
[root@localhost ~]# su tom
[tom@localhost root]$ cd
[tom@localhost ~]$ touch readme.txt
[tom@localhost ~]$ ls -l readme.txt
-rw-rw-r--. 1 tom tom 0 May 27 17:07 readme.txt
[tom@localhost ~]$ chmod u+x readme.txt
[tom@localhost ~]$ ls -l readme.txt
-rwxrw-r--. 1 tom tom 0 May 27 17:07 readme.txt
[tom@localhost ~]$ groups tom
tom : tom wheel google
[tom@localhost ~]$ chmod g+x readme.txt
[tom@localhost ~]$ ls -l readme.txt
-rwxrwxr--. 1 tom tom 0 May 27 17:07 readme.txt
[tom@localhost ~]$ chmod o+t readme.txt
[tom@localhost ~]$ ls -l readme.txt
-rwxrwxr-T. 1 tom tom 0 May 27 17:07 readme.txt
[tom@localhost ~]$ #Have A Good Time All My Facebook Friends (Yin Thu)
[tom@localhost ~]$
```

+ + [Access Control List (ACL)] + +

linux မှာ chmod ဆိုတဲ့ command ကို အသုံးပြုပြီး file တွေ directory တွေကို r(4) w(2) x(1) နဲ့ permission သတ်မှတ်တဲ့အကြောင်းကတော့ ရေးခဲ့ပြီးပါပြီ။

ခုတာခါ တင်ဆက်မှာကတော့ ACL အကြောင်းပါ။ ဟုတ်ပြီ။

ACL ဆိုတာ ဘာလဲ?

ACL ဆိုတာ ပိုပြီး အဆင့်မြင့်တဲ့ permission ပေးခြင်းတစ်မျိုးပါ။

ဥပမာ **dir1** ဆိုတဲ့ directory ရှိတယ် ဆိုပါစို့။

google ဆိုတဲ့ **group** ရှိပြီး **tom** နဲ့ **jerry** ဟာ google **group** ထဲမှာ ပါတဲ့ **user** တွေ ဖြစ်တယ် ဆိုပါစို့။

အဲဒီ google ဆိုတဲ့ **group** ဟာ **dir1** ကို **rwx** နဲ့ access လုပ်ခွင့်ရထားတယ် ဆိုပါစို့။

ဒါဆို google ဆိုတဲ့ **group** ထဲမှာ ပါတဲ့ user အားလုံး (**tom** နဲ့ **jerry**) ဟာ **dir1** ကို **rwx** နဲ့ access လုပ်ခွင့်ရပြီပေါ့။

အဲဒီ အခြေအနေမှာ **tom** ကိုပဲ access လုပ်ခွင့်ပေးမယ်။ **jerry** ကို access လုပ်ခွင့်ပိတ်ထားချင်တယ် ဆိုပါစို့။

ဒီအခါမှာ chmod ဆိုတဲ့ command နဲ့ permission သတ်မှတ်ဖို့ မလုပ်လောက်တော့ပါဘူး။
ဒီနေရာမှာ ACL ဟာ အရေးပါလာပါတယ်။

```
#####
```

ကဲ ! ACL ကို လေ့လာကြည့်ရအောင်။

စမ်းသပ်ဖို့ အတွက် -->

အရင်ဆုံး **tom** နဲ့ **jerry** ဆိုတဲ့ **account** ၂ခု ဆောက်လိုက်ပါ။

```
useradd tom
```

```
passwd tom
```

```
useradd jerry
```

```
passwd jerry
```

google ဆိုတဲ့ **group** တစ်ခု ဆောက်ပါ။ **tom** နဲ့ **jerry** ကို google **group** ထဲ သွင်းပါ။

```
groupadd google
```

```
usermod -aG google tom
```

```
usermod -aG google jerry
```

ပြီးရင် **dir1** နဲ့ **dir2** ဆိုတဲ့ directory လေး ဆောက်လိုက်ပါ။

```
mkdir dir1
```

```
mkdir dir2
```

```
#####
```

file သို့မဟုတ် **directory** တစ်ခုဟာ ACL ကို အသုံးပြုထားမထား ဘယ်လိုကြည့်မလဲ?

file ကို **ls -l** နဲ့ ခေါ်ကြည့်လိုက်လို့ permission အဆုံးမှာ + သက်တပါရင်၏ directory ကို **ls -ld** နဲ့ ခေါ်ကြည့်လိုက်လို့ အဆုံးမှာ + သက်တပါရင်၏ ACL ကို အသုံးပြုထားတယ်လို့ သိနိုင်ပါတယ်။

ဥပမာ **dir1** ကို ACL အသုံးပြုထားမထား ကြည့်မယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **ls -ld dir1**

#####

ACL ကို ဖန်တီးရာမှာ **getfacl** နဲ့ **setfacl** ဆိုတဲ့ command ၂ ခုကို အသုံးပြုရပါတယ်။

၁။ **getfacl** ကို ACL သတ်မှတ်ချက်တွေကို ပြန်ကြည့်ရာမှာ အသုံးပြုရပါတယ်။

ဥပမာ **dir1** ကို သတ်မှတ်ထားတဲ့ ACL ကို ကြည့်မယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **getfacl dir1**

၂။ **setfacl** ကိုတော့ ACL သတ်မှတ်ပေးရာမှာ ပြန်ဖျက်ရာမှာ အသုံးပြုရပါတယ်။

#####

user **tom** ကို **dir1** အတွက် **rwx** ပေးချင်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m u:tom:rwx dir1**

google ဆိုတဲ့ **group** ကို **dir1** အတွက် **rwx** ပေးချင်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m g:googler:rwx dir1**

user **jerry** ကို **dir1** ကို **access** လုပ်ခွင့် ပိတ်ထားချင်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m u:jerry:--- dir1**

#####

user **tom** ကို **default** အဖြစ် သတ်မှတ်ပေးချင်တယ် ဆိုပါလို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m d:u:tom:rwx dir1**

google ဆိုတဲ့ **group** ကို **default** အဖြစ် သတ်မှတ်ပေးချင်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m d:g:hr:rwx dir1**

#####

ACL ထဲက permission တွေကိုမှ **mask** ဆိုတဲ့ option နဲ့ ထပ်ပြီး ကန့်သတ်ချင်တယ် ဆိုပါစွဲ။ ဘယ် user တွေကို ဘာ permission ပေးထားပါစေ **mask** (m) ကို အသုံးပြုပါး **r** လို့ ထပ်မံ ကန့်သတ်လိုက်မယ်ဆိုရင် အဲဒီ file/directory ကို readonly ပဲ လုပ်ခွင့်ရတော့မှာပါ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -m m::r dir1**

#####

ACL သတ်မှတ်ချက်တွေထဲက **tom** နဲ့ဆိုင်တာကို ဖျက်ထုတ်မယ် ဆိုပါစွဲ။ **-x** ဆိုတဲ့ option နဲ့ တဲ့ ပြီး အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -x u:tom dir1**

ACL သတ်မှတ်ချက်တွေထဲက **google** ဆိုတဲ့ **group** ကို ဖျက်ထုတ်မယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ **setfacl -x g:hr dir1**

#####

ACL သတ်မှတ်ချက်တွေ အားလုံးကို ဖျက်ထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။ `setfacl -b dir1`

```
#####
#
```

ACL သတ်မှတ်ချက်တွေထဲက default ပေးထားတဲ့ **user** နဲ့ **group** တွေကိုပဲ ဖျက်ထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ `setfacl -k dir1`

```
#####
#
```

ဟုတ်ပြီ။ `mkdir dir2` ဆိုပြီး directory ၏ ခုထပ်အောက်လိုက်ပါ။

dir1 ကို ACL သတ်မှတ်ပေးထားတယ်။ အဲဒီ သတ်မှတ်ချက်တွေအတိုင်း **dir2** ကို ပုံစံတူ သတ်မှတ်ပေးချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

`getfacl dir1 | setfacl --set-file=- dir2`

```
#####
#
```

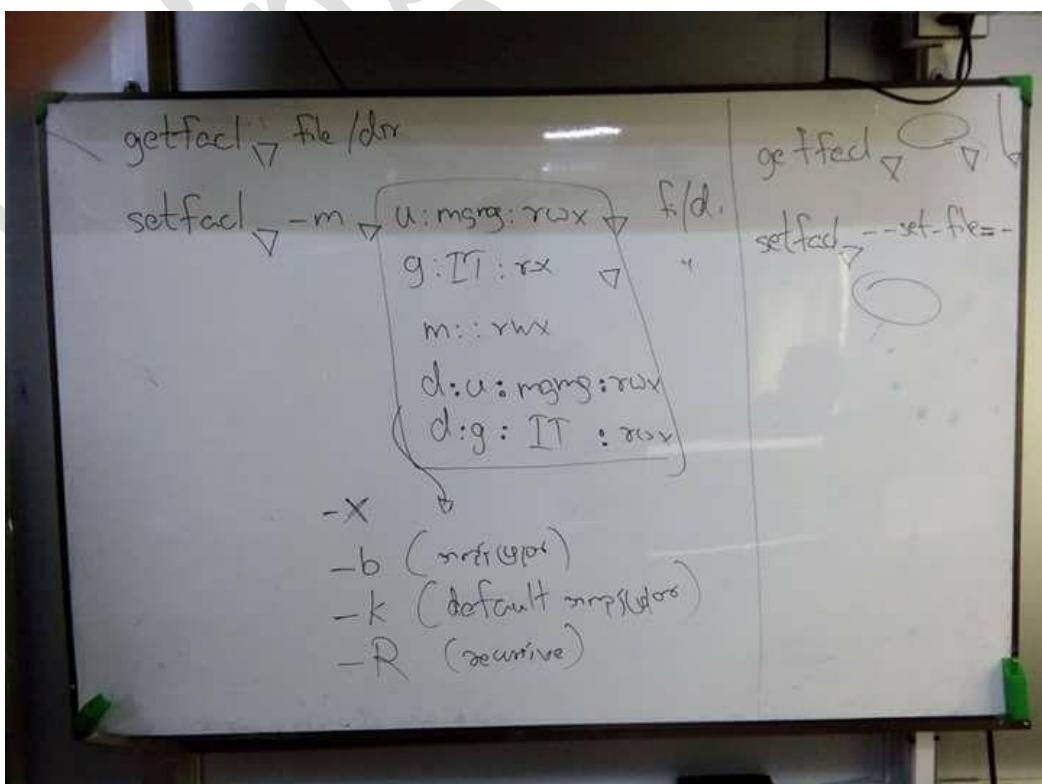
ကဲ ! ACL နဲ့ ပတ်သက်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။ နည်းနည်း ရှုပ်သလို ထင်ရပေမယ့် စမ်းလုပ်သွားရင် အဆင်ပြေသွားပါလိမ့်မယ်။

ACL တစ်ခု သတ်မှတ်ပြီးတိုင်း `getfacl` နဲ့ ပြန်ကြည့် ပြန်စစ်ဖို့လည်း မမေ့ပါနဲ့ `getfacl dir1 << အူလိုမြို့ဗျားပေါ့။`

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(30-5-2018)



default permission and umask

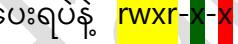
အရမ်း အသုံးနည်းပါတယ်။ ဒါပေမယ့် သိတားသင့်တဲ့ အချက်လေးမို့ပါ။

သတိပြုမယ် ထင်ပါတယ်။ superuser ဖြစ်တဲ့ root account နဲ့ ဖြစ်စေ တစ်ခြား user account တွေ အနေနဲ့ ဖြစ်စေ file တွေ directory တွေကို တည်ဆောက်လိုက်တဲ့ အခါတိုင်း permission တွေ တွဲလျက် ပါလာတတ်တာကိုပေါ့။ linux machine ကို root account နဲ့ login စင်ပါ။

၁။ စမ်းသပ်ဖို့ အတွက် testdir ဆိုတဲ့ directory တစ်ခု ဆောက်လိုက်ပါ။ `mkdir testdir`

ပြီးရင် `ls -ld testdir` နဲ့ ကြည့်ကြည့်ပါ။

၂။ စမ်းသပ်ဖို့ အတွက် `readme.txt` ဆိုတဲ့ file တစ်ခု ဆောက်လိုက်ပါ။ `touch readme.txt`

ပြီးရင် `ls -l readme.txt` နဲ့ ကြည့်ကြည့်ပါ။ ကိုယ်လည်း မပေးရပဲနဲ့  စသည်ဖြင့် permission တွေ ပါလာတာကို တွေ့ရပါလိမ့်မယ်။ အဲဒီ အလိုလို ပါလာတဲ့ permission တွေကို default permission လို့ ခေါ်ပါတယ်။

#default_permission

default permission တွေက ဘာတွေလဲဆိုတာ ကြည့်ရအောင်။

၁။ **root** အနေနဲ့ **directory** ဆောက်ရင် default permission ၡ 755 (rwxr-xr-x)

၂။ **root** အနေနဲ့ **file** ဆောက်ရင် default permission ၡ 644 (rwr--r--)

၃။ **normal user** အနေနဲ့ **directory** ဆောက်ရင် default permission ၡ 775 (rwxrwxr-x)

၄။ normal user အနေနဲ့ file ဆောက်ရင် default permission ၡ 664 (rw-rw-r--)

အဲလို့ တွေ့ရပါမယ်။

#umask

တကယ်တော့ အဲဒီ default permission တွေဟာ umask ဆိုတဲ့ သတ်မှတ် တန်ဖိုးကြောင့်ဖြစ်ပေါ်လာတာပါ။ ဒါကြောင့် ဒီသတ်မှတ်ချက်တွေကို ပြောင်းလဲချင်ရင် umask ရဲ့ value ကို ပြောင်းလဲပေးရပါမယ်။

umask တွေကို ဘယ်မှာ ပြောင်းလဲရမလဲဆိုတော့ `/etc/profile` နဲ့ `/etc/bashrc` ဆိုတဲ့ file ၂ ခု ထဲကို vim text editor နဲ့ ဝင်ရောက် ပြောင်းလဲပေးရပါမယ်။

အဲဒီ file တွေဆီ ဝင်ရောက်ဖို့ အသုံးပြုရမယ့် command ၂ ခုကတော့ အောက်ပါအတိုင်းပါ။

`vi /etc/profile`

`vi /etc/bashrc`

umask ဆိုတဲ့ စကားလုံးကို ရှာပြီး သူဘေးက တန်ဖိုးတွေကို ပြောင်းလဲပေးရမှပါ။

အပေါ်က umask တန်ဖိုးသည် **directory** အတွက် ဖြစ်ပြီး အောက်က umask တန်ဖိုးသည် **file** အတွက် ဖြစ်ပါတယ်။

#maximum_value_for_default_permission

ဒါပေမယ့် အဲဒီလို့၊ မပြောင်းလဲခင် default permission တွေရဲ့ အများဆုံး ခွင့်ပြုထားတဲ့ တန်ဖိုးတွေကို သိထားဖို့ လိုပါတယ်။ အများဆုံး ခွင့်ပြုထားတဲ့ တန်ဖိုးတွေကတော့ -->

directory အတွက်ဆိုရင် 777 (rwxrwxrwx)

file အတွက်ဆိုရင် 666 (rw-rw-rw-)

default permission ကို file အတွက် umask အသုံးပြုပြီး ပြောင်းလဲမယ်ဆို 666 ထက် ပိုပြီး ပြောင်းလဲလို့ မရတဲ့ သဘောပါ။

#FORMULA_to_change_default_permission

ပြောင်းလဲရမယ့် FORMULA ကတော့ အောက်ပါ အတိုင်းပါ။

directory ရဲ့ default permission = (max default value) 777 - umask တန်ဖိုး

file ရဲ့ default permission = (max default value) 666 - umask တန်ဖိုး

ဥပမာ root အနေနဲ့ directory တစ်ခု ဆောက်တိုင်း အလိုလို ပါလာမယ့်

default permission 755 (rwxr-xr-x) ကို 744 (rwxr--r--) အဖြစ် ပြောင်းလဲ သတ်မှတ်မယ် ဆိုပါစို့။ လိုချင်တဲ့ default permission 744 ရတဲ့ အထိ directory အတွက် အများဆုံး ခွင့်ပြုပေးထားတဲ့ permission တန်ဖိုး 777 ထဲကနေ umask တန်ဖိုးကို နှုတ်ပေးရပါမယ်။ အောက်ပါအတိုင်းပေါ့။

လိုချင်သည့် default permission 744 = 777 - 033

ဒါကြောင့် root အနေနဲ့ directory တစ်ခု ဆောက်တိုင်း အလိုလို ပါလာမယ့် default permission

755 (rwxr-xr-x) ကို 744 (rwxr--r--) အဖြစ် ပြောင်းလဲဖို့ အပေါ်ဆုံးမှာ တွေ့ရမယ့် umask တန်ဖိုးကို 033 လို့ ပြောင်းပေးရှုပေါ့။

ဥပမာ root အနေနဲ့ file တစ်ခု ဆောက်တိုင်း အလိုလို ပါလာမယ့် default permission 664 (rw-rw-r) ကို 644 (rw-r--r--) အဖြစ် ပြောင်းလဲ သတ်မှတ်မယ်ဆိုပါစို့။

လိုချင်တဲ့ default permission 644 ရတဲ့ အထိ file အတွက် အများဆုံး ခွင့်ပြုပေးထားတဲ့ permission တန်ဖိုး 666 ထဲကနေ umask တန်ဖိုးကို နှုတ်ပေးရပါမယ်။ အောက်ပါအတိုင်းပေါ့။

လိုချင်သည့် default permission 644 = 666 - 022

ဒါကြောင့် root အနေနဲ့ file တစ်ခု စောက်တိုင်း အလိုလို ပါလာမယ့် default permission 664 (rw-rw-r) ကို 644 (rw-r--r--) အဖြစ် ပြောင်းလဲဖို့ အောက်ဆုံးမှာ တွေ့ရမယ့် umask တန်ဖိုးကို 022 လို့ ပြောင်းပေးရှုပေါ့။

ရှုပ်ထွေးသယောင် ရှိပေါ်မယ့် တကယ်က လွယ်လွယ်လေးပါ။

default permission နဲ့ umask အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(31-5-2018)

```
shum@sol:~ $ ls -l
total 20
drwx--- 2 shum staff 4096 Jan 16 22:04 Mail
drwx--- 3 shum staff 4096 Jan 16 14:15 csc128
drwxr-xr-x 2 shum staff 4096 Jan 13 16:42 public
drwxr-xr-x 2 shum staff 4096 Jan 16 14:07 public_html
-rw-r--r-- 1 shum staff 628 Jan 15 20:04 verse
```

Diagram illustrating the structure of the ls -l output:

- file type:** drwx---
- number of hard links:** 2
- user (owner) name:** shum
- group name:** staff
- other (everyone) permissions:** ---
- group permissions:** r-x
- user permissions:** r-x
- size:** 4096
- filename:** Mail
- date/time last modified:** Jan 16 22:04

RWX

- executable:** r
- writeable:** w
- readable:** x

compression and decompression

ခုတခါ linux ပေါ်မှာ file တွေ directory တွေကို ချုံရာ၊ ပြန်ဖြည့်ရာမှာ အသုံးပြုရတဲ့ command တွေအကြောင်းကိုလေ့လာကြည့်ရအောင်။ ဒီလို ချုံရာ၊ ဖြည့်ရာမှာ tar, zip, gzip စတဲ့ command တွေကို အသုံးပြုကြပါတယ်။

#tar_command

အရင်ဆုံး tar ဆိုတဲ့ command လေးနဲ့ စဖိုင်လိုက်ကြတာပေါ့။ tar ဆိုတဲ့ command ကို အသုံးပြုပြီး ချုံနိုင် ဖြည့်နိုင်တာတွေက အတော်လေး စုပါတယ်။ linux ပေါ်မှာ file တွေ directory တွေကို size သေးသွားအောင် ချုံရဖြည့်ရာမှာ tar ဟာ အသုံးများတဲ့ command တစ်ခုပါ။

#options_for_tar

ဒီလို ချုံရာ၊ ပြန်ဖြည့်ရာမှာ tar နဲ့ အတူ options တွေကို တွဲသုံးရပါတယ်။

tar ရဲ့ options နဲ့ တွဲဘက် အသုံးပြုပုံတွေကို လေ့လာကြည့်ရအောင်။

-c : zip file တွေကို create လုပ်တဲ့အခါ -c (c အသေး) ကို ထည့်ပေးရပါတယ်။

-x : zip file တွေကို ဖြည့်တဲ့အခါ -x ကို ထည့်ပေးရပါတယ်။

-f : သူကတော့ မပါမဖြစ်ပါ။ file တွေကို ချုံတာပဲဖြစ်စေ ပြန်ဖြည့်တာပဲဖြစ်စေ content တွေကို ကြည့်တာပဲဖြစ်စေ -f ကို အမြဲ ထည့်ပေးရပါတယ်။

-p : permission နဲ့ဆိုင်တဲ့ option တစ်ခုပါ။ ချုံရာမှာဖြစ်စေ ပြန်ဖြည့်ရာမှာဖြစ်စေ နဂိုဗုလ permission တန်ဖိုးတွေကို ထိန်းသိမ်းပေးပါတယ်။

-v : ချုံတဲ့ ဖြည့်တဲ့ လုပ်ငန်းစဉ် အခြေအနေကို visual mode နဲ့ ပြေပေးတဲ့ option တစ်ခုပါ။

-t : ချုံထားတဲ့ အထဲက ပါဝင်တဲ့ file တွေကို ကြည့်ဖို့သုံးပါတယ်။

-C : compress လုပ်ထားတဲ့ file ကို တစ်ခြား directory နေရာတစ်ခုမှာ ဖြည့်ချပေးချင်တဲ့အခါ location ကို ညွှန်းဖို့ သုံးပါတယ်။

အထက်မှာ ဖော်ပြထားတာတွေကတော့ **compress/decompress** လုပ်ငန်းစဉ်နဲ့ သက်ဆိုင်တဲ့ option တွေပါ။

#format

compress လုပ်တဲ့ format နဲ့ ဆိုင်တဲ့ option ၁ ငါ မျိုး ရှိပါတယ်။ အောက်ပါအတိုင်းပေါ့။

-z : gz လို့ အတိုကောက် ခေါ်ပြီး **gzip format** နဲ့ ချုံရ ဖြည့်ရာမှာ အသုံးပြုပါတယ်။

ဥပမာ **hacker** ဆိုတဲ့ **directory** ကို **gz** နဲ့ ချုံမယ် ဆိုပါစွဲ။ **command syntax** က အောက်ပါအတိုင်းပါ။ **tar -zcf hacker.gz hacker** (သို့) **tar -zcf hacker.tar.gz hacker** ပြန်ဖြည့်မယ်ဆို၍ **tar -zxf hacker.gz** (သို့) **tar -zxf hacker.tar.gz** ပေါ့။ **tgz** လို့ တွေ့ရင်လည်း **tar.gz** ဆိုတဲ့ **format** နဲ့ ချုံထားတာလို့ သိရပါမယ်။

၂။ **-j : bz2** လို့ အတိုကောက်ခေါ် ပြီး **bzip2 format** နဲ့ ချုံရာ ဖြည့်ရာမှာ အသုံးပြုပါတယ်။
ဥပမာ **hacker** ဆိုတဲ့ **directory** ကို **bz2** နဲ့ ချုံမယ် ဆိုပါစွဲ။ **command syntax** က အောက်ပါ အတိုင်းပါ။ **tar -jcf hacker.bz2 hacker** (သို့) **tar -jcf hacker.tar.bz2 hacker** ပြန်ဖြည့်မယ်ဆို၍ **tar -jxf hacker.bz2** (သို့) **tar -jxf hacker.tar.bz2** ပေါ့။

၃။ **-J : xz** လို့ အတိုကောက်ခေါ် ပြီး **xzip format** နဲ့ ချုံရာ ဖြည့်ရာမှာ အသုံးပြုပါတယ်။
ဥပမာ **hacker** ဆိုတဲ့ **directory** ကို **xz** နဲ့ ချုံမယ် ဆိုပါစွဲ။ **command syntax** က အောက်ပါအတိုင်းပါ။ **tar -Jcf hacker.xz hacker** (သို့) **tar -Jcf hacker.tar.xz hacker** ပြန်ဖြည့်မယ်ဆို၍ **tar -Jxf hacker.xz** (သို့) **tar -Jxf hacker.tar.xz** ပေါ့။

၄။ တစ်ခု ပြောပါရစေ။ ချုံရာမှာ zip format နဲ့ ဆိုင်တဲ့ format တွေ မထည့်ပဲ **-c** နဲ့ **-f** ပဲ သုံးပြီး ချုံမယ် ဆိုပါစွဲ။ ဒီအခါ tar ဆိုတဲ့ format ကို အသုံးပြုပေးရပါလိမ့်မယ်။ အောက်ပါအတိုင်းပေါ့။ **tar -cf hacker.tar hacker** ဒါဆို **tar format** နဲ့ ချုံသွားပါလိမ့်မယ်။ ပြန်ဖြည့်မယ်ဆို **tar -xf hacker.tar** ပေါ့။

#viewing_contents_of_zipfile

zip file ထဲမှာ ပါဝင်တာတွေကို ကြည့်ချင်ရင် **-t** နဲ့ **-f** ကို တွဲသုံးရပါတယ်။
ဥပမာ **hacker.xz** ထဲမှာ ဘာတွေပါလဲ သိချင်တယ် ဆိုပါစွဲ။ **command syntax** က အောက်ပါ အတိုင်းပါ။ **tar -tf gacker.xz**

#extracting_into_other_directory

တစ်ခု ထပ်ပြောပြုပါရစေဦး။ ဥပမာ **current location** **~/** အောက်မှာ ရှိတဲ့ **hacker.xz** ဆိုတဲ့ **zip file** ကို တစ်ခြားတစ်နေရာဖြစ်တဲ့ **/opt/** အောက်မှာ ဖြည့်ချချင်တယ် ဆိုပါစွဲ။ **tar -Jxf hacker.xz /opt/** ဒါလို့ ညွှန်းမယ်ပေါ့။ ဒါဆို error တက်ပါလိမ့်မယ်။

ဒီနေရာမှာ **-C** (**C** အကြံး) ဆိုတဲ့ option က အရေးပါလာပါပြီ။ ဉာဏ်ပေးရမယ့် **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။ **tar -Jxf hacker.xz -C /opt/**
-C အစား **--directory** လိုလည်း သုံးနိုင်ပါတယ်။

#size

file တွေ directory တွေကို ချုပြုလိုက်ပြီး မူလ size ထက် ဘယ်လောက် ပိုကြော့သွားလဲ သိချင်ရင် **du -h** နဲ့ ကြည့်နိုင်ပါတယ်။

ဥပမာ **hacker.xz** ရဲ့ size ကို သိချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **du -h gacker.xz**

ခုလောက်ဆို **tar** နဲ့ options တွေကို လိုအပ်သလို တွဲဖက်ပြီး အသုံးပြုနိုင်မယ် ထင်ပါတယ်။
tar အကြောင်းကတော့ ဒီလောက်ပါပဲ။

#zip_command

zip ဆိုတဲ့ command ကို အသုံးပြုပဲလေး ဆက်ပြောကြတာပေါ့။

zip : zip format နဲ့ ချုမယ်ဆိုရင် **zip -r** ကို သုံးပါတယ်။

ဥပမာ **dir1** ဆိုတဲ့ **directory** ကို **zip format** နဲ့ ချုမယ် ဆိုပါစို့။

အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **zip -r dir1.zip dir1**

ပြန်ဖြည့်မယ်ဆိုရင် **unzip** ကို သုံးပါတယ်။

command က အောက်ပါ အတိုင်းပါ။

unzip dir1.zip dir1

zip ကို အသုံးပြုပြီး ချုံတာ၊ ဖြည့်တာကတော့ ဒီလောက်ပါပဲ။ တို့တို့ရှင်းရှင်းပါပဲနော်။

#gzip_command

gzip အသုံးပြုပဲလေးအကြောင်း ဆက်သွားကြတာပေါ့။

gzip : gzip format နဲ့ ချုမယ်ဆိုရင် **gzip -v** ကို သုံးပါတယ်။ **gzip** ကို အသုံးပြုပြီး **directory**

တွေကို ချုပြု မရပါဘူး။ သူက file တွေပဲ သီးသန့် ချုပြုရပါတယ်။

ဥပမာ **file1** ဆိုတဲ့ **file** ကို **gzip format** နဲ့ ချုမယ် ဆိုပါစို့။

အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **gzip -v file1.gz file1**

ပြန်ဖြည့်မယ်ဆိုရင် **gunzip** ကို သုံးပါတယ်။ command က အောက်ပါအတိုင်းပါ။

gunzip file1.gz

linux ပေါ်မှာ **tar, zip, gzip** စတဲ့ command တွေ အသုံးပြုပြီး file တွေ directory တွေကို ချုပဲဖြည့်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(1-6-2018)



password management

password management ဆိုတာကတော့ password နဲ့ သက်ဆိုင်တာတွေကို စီမံတဲ့ အပိုင်းပါ။ လေ့လာကြည့်ရအောင်။ သိထားရမှာ ၁ ခုကာ password တွေကို manage လုပ်မယ်ဆိုရင် root account နဲ့မှ အဆင်ပြု ပါမယ်။

ကဲ ! linux machine ကို ဖွံ့ဖြိုး root account နဲ့ login ဝင်လိုက်ရအောင်။ password နဲ့ဆိုင်တာတွေ၊ password ပေးခဲ့တဲ့ ရက်စွဲတွေ၊ password တစ်ခု အတွက် အမြင့်ဆုံး သက်တမ်း သတ်မှတ်ချက်တွေ၊ သက်တမ်းကုန်ဆုံးမယ့် expire date,... စတာတွေကို စီမံချင်တဲ့ အခါး **chage** ဆိုတဲ့ command ကို option တွေနဲ့ တွဲသုံးရပါတယ်။

password နဲ့ သက်ဆိုင်တဲ့ အချက် ၂ ချက် ရှုပါတယ်။ အဲဒီ အချက်တွေကို စီမံ သတ်မှတ်ပေးရမှာပါ။ အဲဒါတွေကတော့ အောက်ပါအတိုင်းပါပဲ။

၁။ **Last password change** (password ပေးခဲ့တဲ့ ရက်စွဲ)

၂။ **Password expires** (password သက်တမ်းကုန်မယ့် ရက်စွဲ)

၃။ **Password inactive** (password အသက်ကင်းမဲ့မယ့် နေ့)

၄။ **Account expires** (account သက်တမ်းကုန်မယ့် ရက်စွဲ)

၅။ **Minimum number of days between password change** (password ပြောင်းလဲရမယ့် အနိမ့်ဆုံး ကာလ သတ်မှတ် ရက်ပေါင်း)

၆။ **Maximum number of days between password change** (password ပြောင်းလဲရမယ့် အမြင့်ဆုံး ကာလ သတ်မှတ် ရက်ပေါင်း)

၇။ **Number of days of warning before password expire** (password သက်တမ်း မကုန်ဆုံးမဲ့ ကြိုးတင်သိပေးမယ့် ရက်ပေါင်း)

ဒီနေရာမှာ ရက်စွဲ (date) နဲ့ ရက်ပေါင်း (number of days)၊ မတူတာကို သတိပြုဖို့ မှာပါရစေ။

ကဲ ! command အသုံးပြုပဲလေးတွေ စလိုက်ရအောင်။

password နဲ့ သက်ဆိုင်တဲ့ သတ်မှတ်ချက်တွေကို ကြည့်ချင်တဲ့အခါ । chage -I ကို သုံးပါတယ်။
စောက် ပြောထားတဲ့ အချက် ၂ ချက်ကို မြင်ရပါလိမ့်မယ်။

ဥပမာ tom လို့ အမည်ရတဲ့ user ရဲ့ password ဆိုင်ရာ သတ်မှတ်ချက်တွေကို ကြည့်မယ်ဆိုပါစို့။
command ကတော့ အောက်ပါအတိုင်းပါ။ chage -I tom

၁။ password နောက်ဆုံးတစ်ခေါက် ပြောင်းလဲခဲ့တဲ့ ရက်စွဲ (Last password change) ကို ပြုပြင်း။
chage -d ကို သုံးပါတယ်။ chage -d 2018-06-4 tom

၂။ password သက်တမ်းကုန်မယ့် ရက်စွဲ (Password expires) ကို ပြုပြင်း။ chage -E ကို
သုံးပါတယ်။ date ရဲ့ နောက်မှာ username ထည့်စွဲ မမေ့ပါနဲ့။ chage -E 2018-06-30 tom

၃။ ယခုသူယျအားဖြင့် သက်တမ်းကုန် ပြီးနောက်၊ password အသက်ကင်းမဲ့မယ့် ရက်ပေါင်း
(Password inactive) ဟာ expire date ထက် အမြဲ တန်ဖိုးတစ်ခုကြီးလေ့ ရှိပါတယ်။ password
inactive ဖြစ်မယ့် ရက်ပေါင်းကို ပြောင်းလဲချင်ရင် chage -I (i အကြီး) ကို သုံးတယ်။

chage -I 1 tom

၄။ Account expires ဆိုတာကိုတော့ ဒီနေရာမှာ အထွေအထူး သတ်မှတ်ပေးစရာ မလိုပါဘူး။

၅။ password အသစ် ပြောင်းလဲရမယ့် အနိမ့်ဆုံး ကာလ သတ်မှတ်ချက် (Minimum number of days between password change) ကို ပြုပြင်ဖို့ chage -m ကို သုံးပါတယ်။ ရက်ပေါင်း ခဲ့နောက်မှာ username ထည့်ပေးဖို့ မမေ့ပါနဲ့။ chage -m 20 tom

၆။ password အသစ် ပြောင်းကိုပြောင်းလဲပေးရမယ့် အမြင့်ဆုံး ကာလ သတ်မှတ်ချက် (Maximum number of days between password change) ကို ပြုပြင်ဖို့ chage -M ကို သုံးပါတယ်။ chage -M ခဲ့နောက်မှာ username ထည့်သွင်းပေးဖို့ လိုပါတယ်။

ဥပမာ၊ password နဲ့ သက်ဆိုင်တဲ့ user က tom ဆိုပါစို့ maximum ကာလ ရက်ပေါင်း ၃၀ ပေးမယ်ဆိုပါစို့။ chage -M 30 tom ပေါ့။ chage -M 30 tom

ယော့ယျအားဖြင့်၊ Maximum number of days between password change မှာ တန်ဖိုး ထည့်သွင်းပေးလိုက်ရင် password expire ဖြစ်မယ့် date က အလိုလို သတ်မှတ်ပြီးသား ဖြစ်သွားပါတယ်။ အလားတူ၊ password ခဲ့ inactive ဖြစ်မယ့် date သည်လည်း အလိုလို သတ်မှတ်ပြီးသားဖြစ်သွားပါတယ်။

သိထားသင့်တာ ၁ ခု ကတော့၊ Maximum number of days between password change မှာ default အနေနဲ့ အမြင့်ဆုံး သတ်မှတ်ထားတဲ့ တန်ဖိုးက 99999 ပါ။ အကယ်၍ maximum number of days ကို 99999 လို့ သတ်မှတ်ပေးလိုက်ရင်၊ password expires နဲ့ password inactive ရက်စွဲတွေ နေရာမှာ never လို့ ပြောင်းသွားပါလိမ့်မယ်။ expire မဖြစ်တော့တဲ့ သဘောပေါ့။
chage -M 99999 tom

#calculating_date

ဒီနေရာမှာ maximum ရက်ပေါင်း သတ်မှတ်လိုက်တဲ့ အခါ၊ ခု ဒီနောကာနေ ဘယ်နေ့ထိ အကြံးဝင်မလဲဆိုတဲ့ ရက်စွဲကို date ဆိုတဲ့ command သုံးပြီး တွက်ကြည့်နိုင်ပါတယ်။
ဥပမာ၊ ခုကနေ ရက်ပေါင်း ၃၀ သတ်မှတ်လိုက်တယ် ဆိုပါစို့။ နေ့ရက် တွက်ဖို့ ခုကနေ နောက်လာ မယ့် ရက် ၃၀ ဟာ၊ ဘယ် date ဖြစ်မလဲ သိဖို့ date -d ကို သုံးရုံပါပဲ။ အောက်ပါအတိုင်းပေါ့။

date -d +30days

၇။ password ပြောင်းလဲဖို့ warning message နဲ့ သတိပေးမယ့် ရက်ပေါင်းကို သတ်မှတ်ဖို့ chage -W ကို သုံးပါတယ်။ ရက်ပေါင်း ခဲ့နောက်မှာ username ထည့်ပေးဖို့ မမေ့ပါနဲ့။

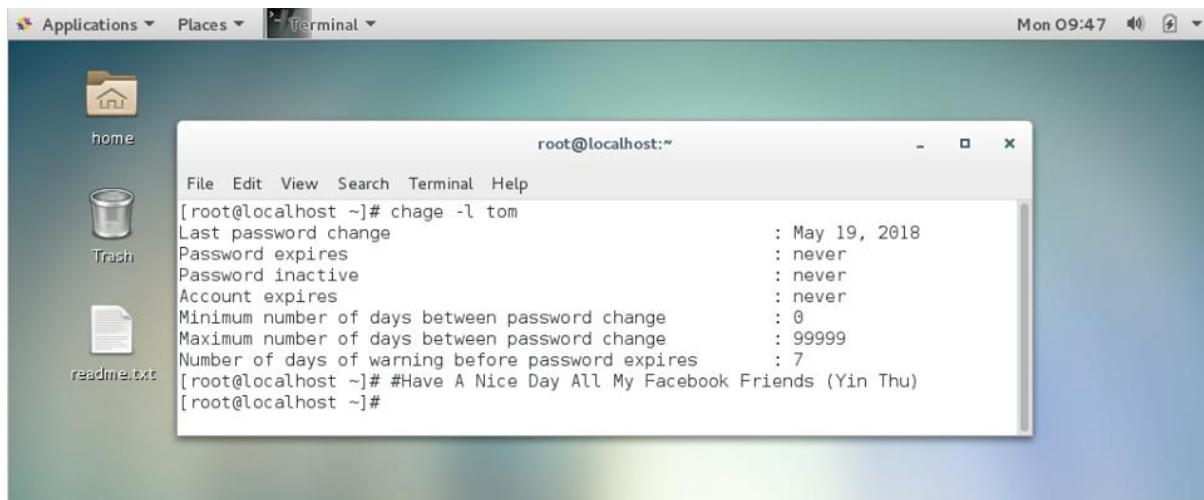
ဥပမာ၊ ၅ ရက် သတ်မှတ်မယ် ဆိုပါစို့။ chage -W 5 username ပေါ့။

chage -W 5 tom

password management နဲ့ သက်ဆိုင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(4-6-2018)



rpm (redhat package manager)

ဒီတစ်ခါတော့ linux ပေါ်မှာ application နဲ့ သက်ဆိုင်တာတွေကို စီမံတဲ့ အခါးပြုတဲ့ rpm command အကြောင်း လေ့လာကြည့်ကြတာပေါ့။ rpm ရဲ့ အရှည်ကောက်က redhat package manager ပါ။ rpm ဟာ red hat, centos, fedora စတဲ့ linux distro တွေမှာ ပါရှုပါတယ်။

မိမိ အတွက် လိုအပ်တဲ့ package တွေကို offline install လုပ်ရာမှာ rpm ကို အများဆုံး အသုံးပြုပါတယ်။

ကဲ ! စလိုက်ရအောင်။

၁။ application package တစ်ခုကို install လုပ်ချင်တဲ့ အခါး rpm -ivh ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ rpm -ivh packagename.rpm အသုံးများတဲ့ command တစ်ခုပါ။
၂။ application တစ်ခုကို ဖယ်ရှားချင်တဲ့ အခါး rpm -e ကို အသုံးပြုပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ rpm -e applicationname

၃။ application တစ်ခုကို update လုပ်ချင်တဲ့ အခါး rpm -Uvh ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ rpm -Uvh applicationname

၄။ application ရဲ့ version ကို ကြည့်ချင်တဲ့ အခါး rpm -q ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ rpm -q applicationname

၅။ application တစ်ခုရဲ့ information တွေကို သိချင်တဲ့အခါ **rpm -qi** ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qi applicationname**

၆။ application နဲ့ သက်ဆိုင်တဲ့ file တွေ ရှိရာ လမ်းကြောင်းတွေကို သိချင်တဲ့အခါ **rpm -ql** ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -ql applicationname**

၇။ application နဲ့ သက်ဆိုင်တဲ့ အဓိက package ကို သိချင်တဲ့အခါ **rpm -qa** ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qa applicationname**

၈။ application နဲ့ သက်ဆိုင်တဲ့ package တွေ အားလုံးကို သိချင်တဲ့အခါ သို့မဟုတ် အဲဒီ application ရှိမရှိ သိချင်တဲ့အခါ **rpm -qa | grep** နဲ့ တွဲသုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qa | grep applicationname** အသုံးများတဲ့ command တစ်ခုပါ။

၉။ file တစ်ခုသည် ဘယ် application နဲ့ သက်ဆိုင်သလဲ၊ သိချင်တဲ့အခါ **rpm -qf** နဲ့ ခေါ်ကြည့် နိုင်ပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qf filename**

၁၀။ application ရဲ့ configuration file ကို သီးသန်း ကြည့်ချင်ရင် **rpm -qc** သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qc applicationname**

၁၁။ application ရဲ့ documentation တွေကို ကြည့်ချင်ရင် **rpm -qd** ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **rpm -qd applicationname**

၁၂။ package တစ်ခုနဲ့တစ်ခုကြား ပြောင်းလဲမှုကို ကြည့်ချင်ရင် **-q --changelog** ကို **rpm** နဲ့ တွဲသုံးရပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။

rpm -q --changelog packagename.rpm

၁၃။ package install လုပ်သွားမယ့် script ကို ကြည့်ချင်ရင် **-q --script** ကို **rpm** နဲ့ တွဲသုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

rpm -q --script packagename.rpm

၁၄။ package ကို မသွင်းပဲ ပါဝင်တဲ့ file ကို ထုတ်ကြည့်ချင်ရင် **rpm2cpio** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ ဥပမာ [sample.rpm](#) ထဲက [readme.txt](#) ဆိုတဲ့ file ကို ထုတ်ကြည့်မယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

rpm2cpio sample.rpm | cpio -id *.txt တိုတိုရှင်းရှင်းပါပဲနော်။ rpm အသုံးပြုဖြီး application တွေကို ကိုင်တွေယ်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(6-6-2018)



yum (yellow-dog update & modify)

linux မှာ application package တွေနဲ့ သက်ဆိုင်တာတွေကို စီမံတဲ့ အခါ debian base ဆိုရင် apt-get, red hat base ဆိုရင် rpm, yum စတဲ့ command တွေကို သုံးပါတယ်။ ဒီနေရာမှာတော့ yum နဲ့ သက်ဆိုင်တဲ့ command တွေကို တင်ဆက်သွားမှာပါ။ yum ရဲအရှည် ကောက်က yellow-dog update and modify ပါ။ online နဲ့ ချိတ်ဆက်ပြီး application တွေကို သွင်းယူရာမှာ yum ကို အများဆုံး အသုံးပြုပါတယ်။ local မှာ ရှိတဲ့ rpm package တွေကိုလည်း yum နဲ့ သွင်းယူနိုင်ပါတယ်။

ကဲ ! စလိုက်ရအောင်။

၁။ offline ဖြစ်တဲ့ မိမိ ရဲ စက်ထဲက application package ကို install လုပ်ချင်တဲ့ အခါ yum -y localinstall ကို အသုံးပြုပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။
yum localinstall -y application.rpm

၂။ online ပေါ်မှာ ရှိတဲ့ application ကို install လုပ်ချင်တဲ့ အခါ yum -y install ကို အသုံးပြုပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ yum -y install applicationname

၁၁။ group လိုက်သွင်းရတဲ့ application တစ်ခုကို install လုပ်ချင်တဲ့အခါ yum groupinstall ကို
သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ yum groupinstall
applicationname

၁၂။ group လိုက်သွင်းထားတဲ့ application တစ်ခုကို ဖယ်ရှားချင်တဲ့အခါ yum groupremove ကို
သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။

yum groupremove applicationname

၁၃။ group လိုက် application တစ်ခုကို update လုပ်ချင်တဲ့အခါ yum groupupdate ကို
သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။

yum groupupdate applicationname

၁၄။ repository ရဲ့စာရင်းကို ကြည့်ချင်တဲ့အခါ yum repolist ကို သုံးပါတယ်။ command syntax
ကတေသာ အောက်ပါအတိုင်းပါ။ yum repolist

၁၅။ enable ပေးထားတဲ့ repo ရော ရော disable ပေးထားတဲ့ repo ရော ကြည့်ချင်တဲ့အခါ yum
repolist all ကို သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ yum repolist all

၁၆။ disable ပေးထားတဲ့ repo ထဲက application တစ်ခုကို install လုပ်ချင်တဲ့အခါ yum --
enablerepo ကို သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။

yum --enablerepo=epel install applicationname

၁၇။ application တစ်ခုကို ရှာဖွေလိုတဲ့အခါ yum search ကို သုံးပါတယ်။ command syntax
ကတေသာ အောက်ပါအတိုင်းပါ။ yum search applicationname

၁၈။ application တစ်ခုရဲ့ information ကို ကြည့်ချင်တဲ့အခါ yum info ကို သုံးပါတယ်။
command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ yum info applicationname

၁၉။ group လိုက် သွင်းရတဲ့ application group တစ်ခုရဲ့ information ကို ကြည့်ချင်တဲ့အခါ yum
groupinfo ကို သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။

yum groupinfo "applicationgroupname"

၂၀။ yum နဲ့ သက်ဆိုင်တဲ့ history ကို ကြည့်ချင်တဲ့အခါ yum history ကို သုံးပါတယ်။ command
syntax ကတေသာ အောက်ပါအတိုင်းပါ။ yum history

၂၁။ file တစ်ခုဟာ ဘယ် application နဲ့ သက်ဆိုင်သလဲ ကြည့်ချင်ရင် yum provides ကို
သုံးပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ yum provides /usr/filename

၂၂။ history ကို ရှင်းလင်းချင်တဲ့အခါ yum clean ကို သုံးပါတယ်။ command syntax ကတေသာ
အောက်ပါအတိုင်းပါ။ yum clean all

၂၃။ yum အတွက် custom shell ကို ဖွင့်ချင်တဲ့အခါ yum shell ကို သုံးပါတယ်။ command syntax
ကတေသာ အောက်ပါအတိုင်းပါ။ yum shell

ခုလောက်ဆို yum ကို အသုံးပြုပြီး application နဲ့ သက်ဆိုင်တာတွေကို ကိုင်တွယ်နိုင်ပြီ ထင်ပါတယ်။ **yum** နဲ့ သက်ဆိုင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(6-6-2018)



rpm Vs yum

red hat, centos, fedora linux စတာတွေရဲ့ application နဲ့ ဆိုင်တဲ့ အပိုင်းတွေကို စီမံရှာမှာ rpm, yum စတဲ့ command တွေကို အသုံးပြုတဲ့ အကြောင်း တင်ဆက်ပြီးသွားပါပြီ။
အခုတစ်ခါ ဒီ command နှစ်ခုရဲ့ အားသာချက်၊ အားနည်းချက်တွေကို ယဉ်ကျဉ်းကြဖို့။

#rpm

ဥပမာ A >> စက်မှာ **zimbra mail server** ကို rpm command အသုံးပြုပြီး install လုပ်မယ် ဆိုပါစို့။ rpm -ivh zimbra....rpm ပေါ့။ အဲဒီအခါ perl package လိုနေပါတယ်ဆိုပြီး ဆက်သွင်းလို့ မရတော့ပဲ error ပြပါလိမ့်မယ်။ ဟုတ်ပြီ perl package သွင်းပြီးသွားပြီဆိုပါစို့။ တစ်ခါ **zimbra** ကို အစကနေ rpm နဲ့ ပြန်သွင်း။ rpm -ivh zimbra....rpm ပေါ့။ အဲဒီမှာ **sysstat package** လိုနေတယ် ဆိုပြီး ထပ် error ပြပြန်ရော့။ sysstat ကို အရင်ထပ်သွင်းပြီးမှ **zimbra** ကို အစကနေ rpm နဲ့ ပြန်သွင်းရပြန်ရော့။ rpm -ivh zimbra....rpm ပေါ့။ \$ရမ်း အာရုံနောက်ပါတယ်နော်။

ကဲ ! rpm အစား yum ကို အသုံးပြုကြည့်ရအောင်။

#yum

ဥပမာ B >> yum localinstall zimbra....rpm ပေါ့။

ဒီနေရာမှာ yum ဟာ လိုအပ်နေတဲ့ perl နဲ့ sysstat package တွေကို ဘာ error မှ မပြပဲ၊ အလိုလျောက် သွင်းယူပြီး zimbra ကို သွင်းပေးသွားပါလိမ့်မယ်။ online ဖြစ်နေဖို့တော့ လိုတာပေါ့။ ဘယ်လောက် အာရုံနောက် သက်သာလဲ? ခုလောက်ဆို rpm နဲ့ yum ဘယ်ဟာပိုလန်းသလဲ သိလောက်ပါပြီနော်။ ဒါပေမယ့် သူနေရာနဲ့သူပေါ့နော်။ ကာလံအေသံ အလိုက် သုံးစွဲနိုင်ကြပါစေ။

အားလုံး အဆင်ပြုပါစေ။

Redhat Package Management



Interview Questions

adding repo link

linux မှာ **yum**, **rpm** စတဲ့ command တွေကို သုံးပြီး application တွေ install, update လုပ်နိုင်ဖြုံး
repo link တွေ ရှိမှ အဆင်ပြေပါတယ်။ အဲဒီ **repo link** တွေကို **/etc/yum.repos.d/** အောက်မှာ
.repo နဲ့ ဆုံးတဲ့ **file** တွေ အဖြစ် တွေ့ရှိနိုင်ပါတယ်။ **cd /etc/yum.repos.d/ ls**
လိုအပ်ချက်အရ **repo link** တွေ ထပ်မံထည့်သွင်းလိုတဲ့အခါ **/etc/yum.repos.d/** အောက်မှာ
vim command line text editor နဲ့ **.repo** နဲ့ အဆုံးသတ်တဲ့ **file** တွေ အဖြစ် သွားစောက်ပေး
ရပါတယ်။ **file** အမည်ကို နှစ်သက်သလို ပေးနိုင်ပြီး extension က **.repo** နဲ့ ဆုံးဖို့ လိုပါတယ်။

#adding_repo_link_manually

repo file တစ်ခုထဲမှာ အောက်ပါအချက်တွေ ပါဝင်ရပါမယ်။

[repo id]

name=my repo

baseurl="http://dl.fedoraproject.org/pub/epel/7/x86_x64/"

enabled=1

gpgcheck=0

ရှင်းရရင် --> [...] ထဲက အရာကို **repo id** လို့ ခေါ်ပါတယ်။ နှစ်သက်သလို ပေးနိုင်ပါတယ်။ မေ့မ
သွားဖို့ပဲလိုတာ ပါ။

name နေရာမှာ မိမိ နှစ်သက်သလို ရေးနိုင်ပါတယ်။ မရေးပဲ လွှတ်ထားခဲ့လည်း အဆင်ပြေပါတယ်။

baseurl ကတော့ အရေးအကြီးဆုံးပါ။ သွားရောက် ချိတ်ဆက်မယ့် link ကို တိတိကျကျ ထည့်သွင်း
ပေးဖို့ လိုပါတယ်။

enabled ဆိုတာကတော့ **repo** ကို **enable** ပေးမလား **disable** ပေးမလား သတ်မှတ်တဲ့ အပိုင်းပါ။

1 ဆိုရင် **enable** ဖြစ်ပြီး **0** ဆိုရင် **disable** လို့ သတ်မှတ်ပါတယ်။

gpgcheck ဆိုတာကတော့ **gpg key** ထည့်မထည့် သတ်မှတ်တဲ့ အပိုင်းပါ။ **1** ဆိုရင် **gpgkey**
ထည့်မယ် လို့ သတ်မှတ်ပြီး **0** ဆိုရင် မထည့်ဘူးလို့ သတ်မှတ်ပါတယ်။ ဒါတွေကတော့ **repo file**
တစ်ခုကို manual တည်ဆောက်တဲ့ အပိုင်းပါ။

#adding_repo_link_via_command

command line ကနေ တည်ဆောက်ချင်ရင် **yum-config-manager** ဆိုတဲ့ command ကို
သုံးရပါတယ်။ ဥပမာ **repo file** တစ်ခု စောက်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က
အောက်ပါအတိုင်းပါ။

yum-config-manager --add-repo http://dl.fedoraproject.org/pub/epel/7/x86_x64/

#enable

repo တစ်ခုကို enable ပေးမယ် ဆိုပါစွဲ။ command syntax က အောက်ပါအတိုင်းပါ။

yum-config-manager --enable repo_id

repo id နေရာမှာ **repo file** ထဲက [...] ထဲမှာ ရှိတဲ့ အရာကို ရေးပေးရမှာပါ။ file name ကို ရေးပေးရမှာ မဟုတ်ပါဘူး။

#disable

repo တစ်ခုကို **disable** ပေးမယ် ဆိုပါစွဲ။ **disable** ပေးတယ်ဆိုကတည်းက **yum** နဲ့ **application** တွေ install လုပ်တဲ့အခါ ဒီ **repo** ကို မသုံးဘူးဆိုတဲ့ သဘောပါပဲ။ command syntax က အောက်ပါအတိုင်းပါ။ **yum-config-manager --disable repo_id**

#checking

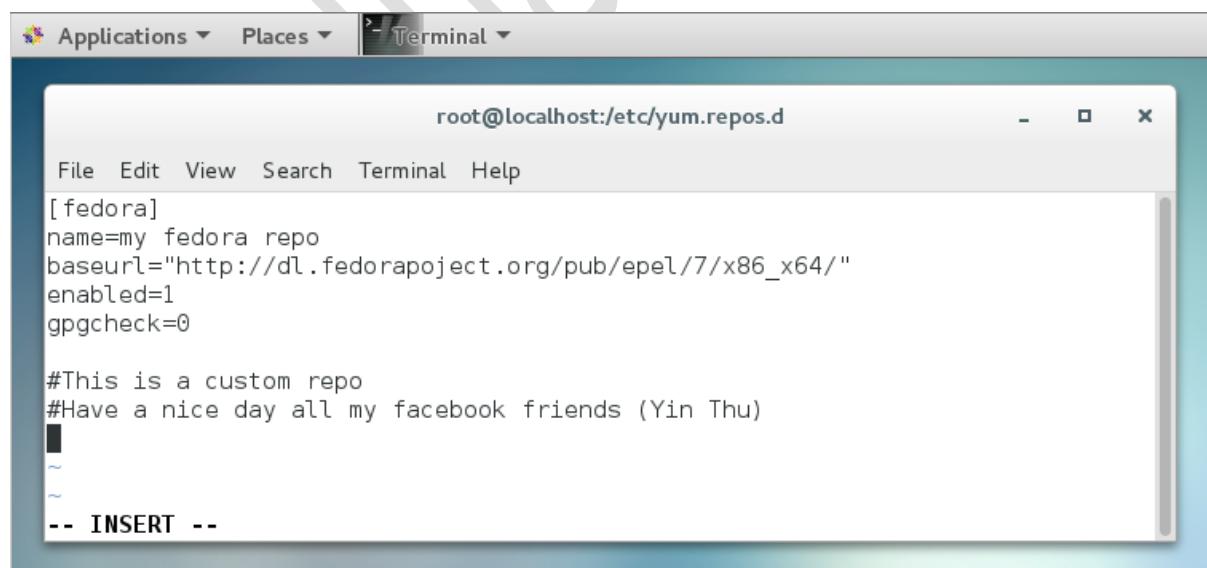
repo list တွေ သူတို့ရဲ့ **repo_id** တွေကို yum command နဲ့ ခေါ်ကြည့်နိုင်ပါတယ်။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **yum repolist all**

all မထည့်ပဲ **yum repolist** နဲ့ ခေါ်ကြည့်ရင်တော့ **enable** ပေးထားတဲ့ **list** တွေကိုပဲ တွေ့ရပါလိမ့်မယ်။ **yum repolist**

#####
#####

linux ပေါ်မှာ application တွေကို install, update လုပ်နိုင်ဖို့လိုအပ်ချက်အရ repo file တွေ ထပ်မံတည်ဆောက်တဲ့အပိုင်း၊ enable or disable ပေးတဲ့အပိုင်း၊ **ပြန်ကြည့်တဲ့အပိုင်း**ကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu (8-6-2018)



```
root@localhost:/etc/yum.repos.d
File Edit View Search Terminal Help
[ fedora ]
name=my fedora repo
baseurl="http://dl.fedoraproject.org/pub/epel/7/x86_64/"
enabled=1
gpgcheck=0

#This is a custom repo
#Have a nice day all my facebook friends (Yin Thu)

-- INSERT --
```

controlling services (services များကို စီမံခွင့်)

linux ပေါ်မှာ run နေတဲ့ services တွေကို ကိုင်တွယ်ဖို့ systemctl ဆိုတဲ့ command ကို သုံးပါတယ်။ application တွေကို unit လို့ ခေါ် ပြီး၊ unit တစ်ခုမှာ

၁။ service

၂။ socket

၃။ path ဆိုပြီး ၃ ပိုင်းရှိပါတယ်။

၁။ active ဖြစ်နေတဲ့ units တွေရဲ့ service ကို ကြည့်ချင်တဲ့အခါ systemctl list-units --type=service ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl list-units --type=service

၂။ inactive ဖြစ်နေတဲ့ units တွေရဲ့ service ကို ကြည့်ချင်တဲ့အခါ systemctl list-units --type=service --all --state=inactive ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl list-units --type=service --all --state=inactive

၃။ active ဖြစ်နေတဲ့ units တွေရဲ့ service ရော့ inactive ဖြစ်နေတဲ့ units တွေရဲ့ service ကိုပါ ကြည့်ချင်တဲ့အခါ systemctl list-units --type=service --all ကိုသုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl list-units --type=service --all

၄။ units တွေရဲ့ services တွေ enable ဖြစ်နေလား disabled ဖြစ်နေလား သိဖို့ systemctl list-unit-files --type=service ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl list-unit-files --type=service

၅။ failed ဖြစ်နေတဲ့ units တွေကို ကြည့်ဖို့ systemctl --failed --type=service ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl --failed --type=service

၆။ unit တစ်ခုရဲ့ status ကို ကြည့်ဖို့ systemctl status UNIT.service ကို သုံးတယ်။ UNIT နေရာမှာ application ကို ထည့်ပေးရမှာပါ။

ဥပမာ sshd ဆိုတဲ့ service ကို ကြည့်မယ် ဆိုပါစွဲ။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl status sshd.service

၇။ unit တစ်ခုရဲ့ status:full output ကို ကြည့်ချင်ရင် systemctl status sshd.service -l ကို သုံးတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl status sshd.service -l

၈။ unit တစ်ခု active ဖြစ်နေလား သိဖို့ systemctl is-active UNIT ကို သုံးပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။
systemctl is-active sshd

၉။ **unit** ထစ်ခု **boot time** မှာ **run** ဖို့ (စက်စတက်တက်ချင်း run ဖို့) **enable** ပေးခံထားရလား
သိဖို့ `systemctl is-enabled UNIT` ဆိုတဲ့ **format** ကိုသုံးပါတယ်။ command ကတော့
အောက်ပါအတိုင်းပါ။ `systemctl is-enabled sshd`

ခုလိုမျိုး အလားတူ၊ Systemctl နဲ့ တွဲဖက် အသုံးပြုနိုင်တဲ့ command statement ဘု ခုကို
အတိုချုပ် ဖော်ပြပေးလိုက်ပါတယ်။ UNIT နေရာမှာ စီမံလိုတဲ့ application ကို ထည့်ပေးရမှာပါ။
ရိုးရိုးရှင်းရှင်းပါပဲနော်။

Summary of systemctl commands

1.view detailed information about a unit state

`systemctl status UNIT`

2.start a service on a running system s

`systemctl start UNIT`

3.stop a service on a running system

`systemctl stop UNIT`

4.restart a service on a running system

`systemctl restart UNIT`

5.reload configuration file of a running service

`systemctl reload UNIT`

6.completely disable a service from being start

`systemctl mask UNIT`

7.make a mask service available

`systemctl unmask UNIT`

8.configure a service to start at boot time

`systemctl enable UNIT`

9.disable a service from starting at boot time

`systemctl disable UNIT`

10.to see the low-level properties of a unit

`systemctl show UNIT`

11.list units which are required by the specified unit

`systemctl list-dependencies UNIT`

12.to halt the system

systemctl halt

13.to shutdown the system

systemctl poweroff

14.to reboot the system

systemctl reboot

15. to hibernate the system

systemctl hibernate

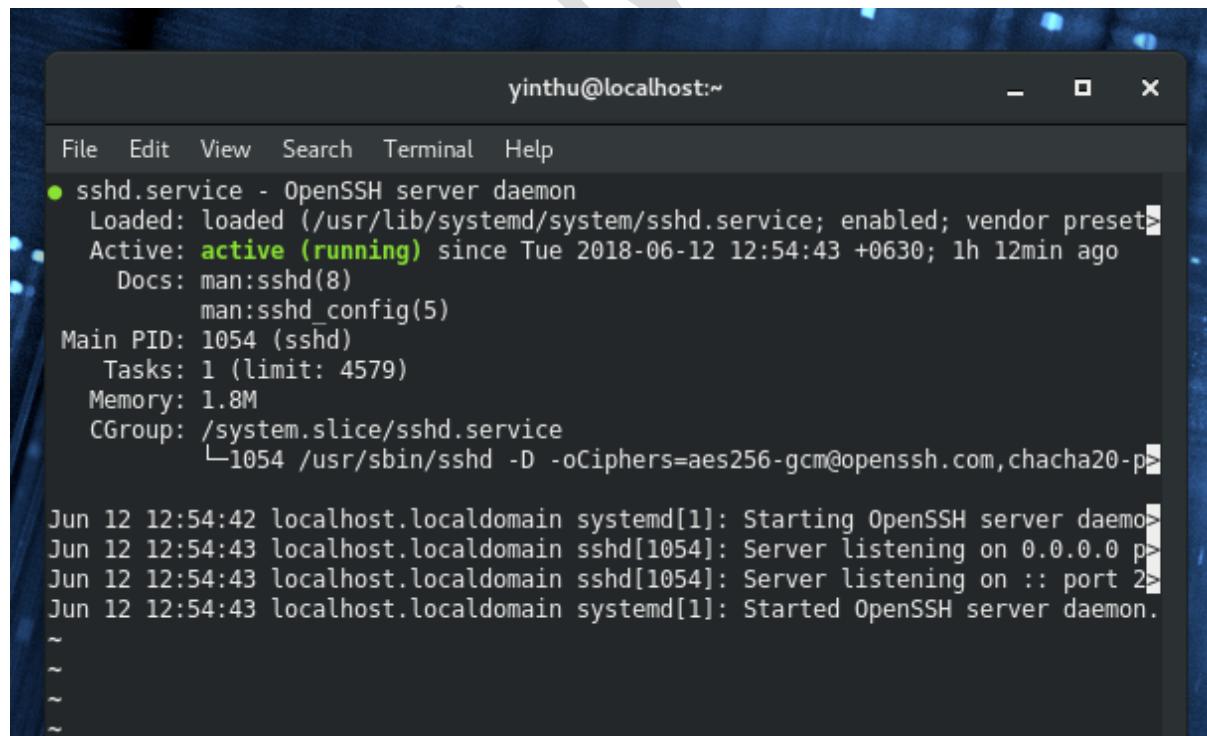
linux 7 မတိုင်ခ် old version တွေမှာတော့ systemctl အစား service ဆိုတဲ့ command ကို
သုံးပါတယ်။ သဘောတရားချင်းကတော့ အတူတူပါပဲ။ service ကို အသုံးပြုမယ်ဆို သူ့ရဲ့ formant
က **service applicationcommand statement** ပါ။

ဥပမာ old version မှာ sshd ကို restart လုပ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command က
အောက်ပါအတိုင်းပါ။ **service sshd restart**

ကဲ ! systemctl ကို အသုံးပြုပြီး service တွေကို ကိုင်တွယ်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(12-6-2018)



yinthu@localhost:~

```
File Edit View Search Terminal Help
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2018-06-12 12:54:43 +0630; 1h 12min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1054 (sshd)
     Tasks: 1 (limit: 4579)
    Memory: 1.8M
   CGroup: /system.slice/sshd.service
           └─1054 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,ecdh-sha2-nistp256@openssh.com,ecdh-sha2-nistp384@openssh.com,ecdh-sha2-nistp521@openssh.com,rsa-sha2-256@openssh.com,rsa-sha2-512@openssh.com,dss-sha2-256@openssh.com,dss-sha2-512@openssh.com

Jun 12 12:54:42 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jun 12 12:54:43 localhost.localdomain sshd[1054]: Server listening on 0.0.0.0 port 22...
Jun 12 12:54:43 localhost.localdomain sshd[1054]: Server listening on :: port 22...
Jun 12 12:54:43 localhost.localdomain systemd[1]: Started OpenSSH server daemon.

~
```

(ssh အပိုင်း ၁)

ssh ဖြင့် password ကို အခြေပြု၍ remote login ဝင်ခြင်း

ဒီတစ်ခါ linux machine တွေကို secure shell လို့ ခေါ်တဲ့ ssh နဲ့ remote login ဝင်နည်းလေးကို လေ့လာရအောင်များ။

တကယ့်ပါးပါးလေး လေ့လာကြည့်ကြတာပေါ့။

ဥပမာ - server ဆိုတဲ့စက်ကို client ဆိုတဲ့ စက်ကနေ၊ remote login ဝင်မယ် ဆိုပါစို့။ server ပေါ်မှာ ကြိုက်တည်ဆောက်ထားတဲ့ account ရှိဖို့လိုပါတယ်။ ဒါမှ အားဖြီ username နဲ့ password ကို သုံးပြီး remote login ဝင်လို့ရမှာပါ။

#requirement

ssh နဲ့ မဝင်ခင် **sshd** ဆိုတဲ့ **service** ကို စက်မှာ run ထားဖို့ လိုပါတယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။ **systemctl start sshd** (or) **systemctl enable sshd**

#log_in_with_hostname

ဒီလို့ log in ဝင်ဖို့ server ရဲ့ hostname ကိုလည်း သိဖို့ လိုပါတယ်။ server ရဲ့ hostname ကို သိဖို့ server ပေါ်မှာ hostname ဆိုတဲ့ command ကို run ပါး ကြည့်နိုင်ပါတယ်။ **www.example.com** လို့ ဆိုကြပါစို့။ server ပေါ်မှာ **root** ဆိုတဲ့ account လေး ရှိတယ်ဆိုပါစို့။ သူ့ကို remote login ဝင်ဖို့က အရမ်းကို လွယ်ပါတယ်။ ssh ဆိုတဲ့ command လေးနဲ့ တွဲပြီး အောက်ပါအတိုင်း ရှိက်ပေးရုပါပဲ။ **ssh root@www.server.example.com** yes or no မေးရင် yes လို့ ရှိက် enter ခေါက်ပါ။ password တောင်းပါလိမ့်မယ်။ ထည့်ပေးလိုက်ပါ။ ဒါခုံဗုရင် **server** ရဲ့ command line ထဲကို remote login နဲ့ ဝင်ရောက်သွားပြီပေါ့။ ပြန်ထွက်မယ်ဆို **exit** လို့ ရှိက် enter ခေါက်ပြီး ပြန်ထွက်ပါ။

#log_in_with_ip

server ရဲ့ **hostname** ကို မသိလည်း ပြသနာ မရှိပါဘူး။ **ip address** ကို သိရင် အဆင်ပြေပါတယ်။ ဥပမာ - **server** ရဲ့ **ip** က **127.0.0.1** ဆိုပါစို့။ **ssh root@127.0.0.1** ပေါ့။ ပြန်ထွက်မယ်ဆို **exit** လို့ ရှိက် enter ခေါက်ပြီး ပြန်ထွက်ပါ။

#remark

တစ်ချက် သိထားဖို့က ssh ကို အသုံးပြုတဲ့အခါ **server** ကနေ **client** ဆိုကို **encrypted** လုပ်ထားတဲ့ **public key** တစ်ခုကို ပို့ပေးပါတယ်။ **~/.ssh/known_host** ဆိုတဲ့ file လေးပေါ့။ သူ့ကို ကြည့်ချင်ရင် --> **cd ~/.ssh** **ls -alt** နဲ့ ကြည့်နိုင်ပါတယ်။ server ဘက်ခြမ်းမှာတော့ **/etc/ssh/*key*** အောက်မှာ သိမ်းထားပါတယ်။ **server** ကို ssh နဲ့ **login** ဝင်တိုင်း အဲဒီ **key** နှစ်ခုကို တိုက်ဆိုင် စစ်ဆေးပါတယ်။ **server** ဘက်က ပြောင်းလိုက်လိုဖြစ်စေ အကြောင်းတစ်ခုကြောင့်

ဖြစ်စေ **key pair** မတူတော့ရင် ဝင်လို့ မရတော့ပါဘူး။ အဲဒီအခါ **client** ဘက်မှာ ရှိတဲ့ [~/.ssh/known_host](#) ဆိုတဲ့ **file** ကို ဖျက်ပစ်ပါး **ssh** နဲ့ ထပ်မံ **login** ဝင်ပေးဖို့ လိုပါတယ်။

rm -rf ~/.ssh/known_host ဒါကတော့ သိထားဖိုပါ။ လွယ်ပါတယ်နော်။ **ssh** အပိုင်း ၁ ကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(13-6-2018)

```
[yinthu@localhost ~]$ ssh root@www.example.com
The authenticity of host 'www.example.com (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:+gj6+XVg44gc0I5ciGCri9xoElBAAgWOFgAYvZ1V7Ts.
ECDSA key fingerprint is MD5:76:ce:6c:3d:47:1b:4e:1b:a3:ae:db:06:3b:0d:5f:3b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'www.example.com' (ECDSA) to the list of known hosts.
root@www.example.com's password:
Last login: Tue Jun 12 23:25:17 2018
[root@localhost ~]# exit
logout
Connection to www.example.com closed.
[yinthu@localhost ~]$
[yinthu@localhost ~]$ # Have A Nice Day All My Facebook Friends (Yin Thu)
```

(ssh အပိုင်း၂) ssh with private-public key

linux machine ကို ssh ကနေတစ်ဆင့် password နဲ့ remote login ဝင်တာကိုတော့ အပိုင်း ၁ မှာ အာလုံး ဖုတ်ပြီးသွားပါပြီ။ ဒီတစ်ခါ password သုံးစရာမလိုပဲ private-public key အသုံးပြု ပြီး ဝင်နည်းကို လေ့လာကြည့်ရအောင်။ ဒီနည်းက ပိုပြီး လုံခြုံမှု ရှိပါတယ်။

ဥပမာ >> client ဆိုတဲ့ စက်ကနေ server ပေါ်မှာ ရှိတဲ့ root ဆိုတဲ့ account ဆီ remote login ဝင်မယ် ဆိုကြပါစို့။ လုပ်ရမယ့် အဆင့်တွေက ရှိုးရှင်းပါတယ်။ client စက်မှာ private key နဲ့ public key နှစ်ခုကို ဖန်တီးမယ်။ ပြီးရင် server စက်ဆီ public key ကို လှမ်းပို့မယ်။ တစ်ခါပြုပြီးသွားရင် နောက်တစ်ခါ remote login ဝင်တဲ့အခါ ဘာမှ password ထည့်စရာ မလို တော့ဘူးပေါ့။ ဒီလောက်လေးပါပဲ။

ကဲ ! client စက်မှာ private နဲ့ public key နှစ်ခုကို ဖန်တီးဖို့ အောက်က command လေး run လိုက်ကြဖို့။ **ssh-keygen** ဘာဘာညာညာ ဖြည့်ခိုင်းရင် enter ပဲ ခေါက်သွားလိုက်ပါ။

ဒီလို run ပြီးတဲ့အခါ [~/.ssh/](#) အောက်မှာ private key ကို **id_rsa** နဲ့ **public key** ကို **id_rsa.pub** ဆိုပြီး တွေ့ရပါမယ်။ key တွေ ဖန်တီးပြီးသွားပြီဆိုတော့ public key ကို server ဆီ command လေးနဲ့ အောက်ကအတိုင်း ပို့ပေးလိုက်ရုံပေါ့။

ssh-copy-id -i ~/.ssh/id_rsa.pub root@www.example.com

ဒါဆိုရင် လုပ်ရမယ့် အဆင့်တွေ ပြီးဆုံးပါပြီ။ server ပေါ်က root ဆိုတဲ့ account ဆီ ssh နဲ့ login ဝင်ကြည့်လိုက်ပါ။ **ssh root@www.example.com** ဘာ password မှ မတောင်းတော့တာ တွေ့ရပါ လိမ့်မယ်။

ssh အပိုင်း ၂ ကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(14-6-2018)

```
File Edit View Terminal Help
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yinthu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/yinthu/.ssh/id_rsa.
Your public key has been saved in /home/yinthu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:cIfhsY8F9vPpsyuLqtqdVegNZrWRdpJ8Qyx/YvVPhg8 yinthu@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]---+
|      = .. |
|     o 0+. . |
|    . = &0+. o |
|   o 0 0+oE +|
|    S +.oo =.|
|   ++ .     o |
|    o . o    |
|   . . o ... o |
|  ..o.+...oo. |
+---[SHA256]---+
[yinthu@localhost ~]$ cd .ssh
[yinthu@localhost .ssh]$ ls
id_rsa  id_rsa.pub  known_hosts
[yinthu@localhost .ssh]$
```

(ssh အပိုင်း ၃)

root account ဆီသို့ ssh ဖြင့် remote login ဝင်ခြင်းမှ တားဆီးခြင်း ##

ဒီတစ်ခါတော့ linux server ပေါ်က root account ဆီကို ssh သုံးပြီး remote login ဝင်လို့
မရအောင် ပိတ်ထားကြတို့။ ဘာ့ကြောင့်လဲ ဆိုတော့ linux machine တစ်ခု မှာ root account ဟာ
အရေးအပါဆုံးမွှေ့ သုကို ssh နဲ့ login ဝင်ခွင့် ပေးထားတာဟာ အတန်အသင့် အန္တရာယ်ကြီးလှ
ပါတယ်။ ဒါ့ကြောင့်ပါ။ လုပ်ရမယ့် အဆင့်တွေကတော့ ရှိုးရှင်းပါတယ်။

server စက်ထဲက /etc/ssh/sshd_config ဆိုတဲ့ file ထဲကို ဝင်ပြင်ပေးရုပါပဲ။ command ကတော့
အောက်ပါအတိုင်းပေါ့။ **vi /etc/ssh/sshd_config** i key ကို နှိပ်ပြီး edit mode ထဲ ဝင်ပါ။ ဘာတွေ
ပြင်ရမလဲဆိုတော့ **PermitRootLogin yes** နေရာမှာ **no** ... အားလုံး ပြင်ပေးလိုက်ပါ။ ပြီးတဲ့အခါ

escape key နှင်း :wq ရှိက်ပြီး ထွက်ပါ။ ဒါအပြင် အောက်က command လေးနဲ့ sshd service ကို restart ချလိုက်ပါဉိုး။ **service sshd restart** ဒါမှ ပြင်ဆင်ထားတာတွေ အသက်ဝင်မှာပါ။ ကဲ ! client စက်ဆီသွား၊ server ရဲ့ root account ဆီကို ssh နဲ့ remote login ဝင်ကြည့်ပါဉိုး။ **ssh root@www.example.com** ဝင်လို့ ရပါသေးရဲ့လား?

မိမိ server ရဲ့ root account ဆီကို ssh နဲ့ ဝင်မရအောင် တားဆီးတာကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(14-6-2018)

```
# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes <-- change yes to no (Yin Thu)
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
-- INSERT --
```

(ssh အပိုင်း င) normal account အား ssh ဖြင့် remote login ဝင်ခြင်းမှ တားဆီးခြင်း ##
root account ကို တားဆီးသလိုပါပဲ။ server စက်ထဲက /etc/ssh/sshd_config ဆိုတဲ့ file ထဲကို ဝင်ပြင်ပေးရမှာပါ။ command ကတော့ အောက်ပါအတိုင်းပေါ့။ vi /etc/ssh/sshd_config i key
ကို နှင်းပြီး **edit mode** ထဲ ဝင်ပါ။ ဘာတွေပြင်ရမလဲဆိုတော့ **PasswordAuthentication yes** နေရာမှာ **no** အားလုံး ပြင်ပေးလိုက်ပါ။ ပြီးတဲ့အခါ **escape key** နှင်း :wq ရှိက်ပြီး ထွက်ပါ။ ဒါအပြင်

အောက်က command လေးနဲ့ sshd service ကို restart ချလိုက်ပါ။ `systemctl restart sshd` ဒါမှ ပြင်ဆင်ထားတာတွေ အသက်ဝင်မှာပါ။

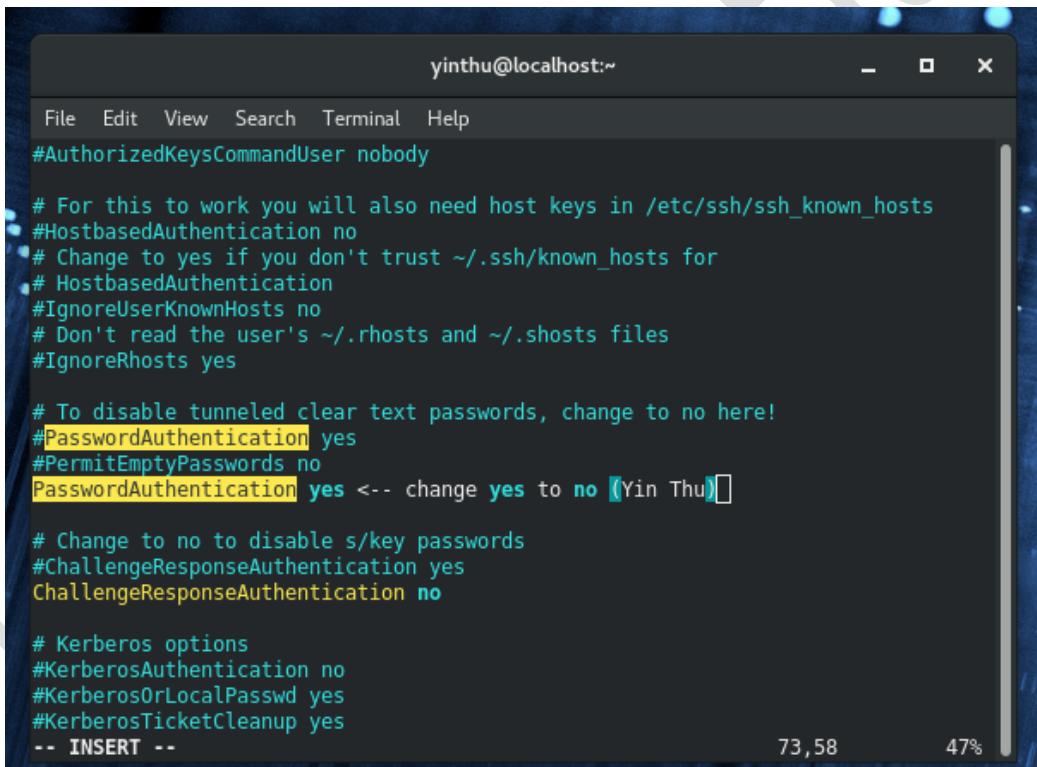
ကဲ ! client စက်ဆီသွား၊ server ပေါ်က normal account ဆီကို ssh နဲ့ remote login ဝင်ကြည့်ပါ။ `ssh normaluser@example.com` ဝင်လို့ ရပါသေးရဲ့လား?

ဒီနေရာမှာ တစ်ခု သိထားရမှာက၊ ဒီလို့ မတားဆီးခင် user account တွေဟာ (**Private-Public key-based authentication**) ကြိုလုပ်ထားမယ်ဆိုရင်၊ remote login ဝင်လို့ ရနေမယ် ဆိုတာပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(15-6-2018)



```
yinthu@localhost:~  
File Edit View Search Terminal Help  
#AuthorizedKeysCommandUser nobody  
  
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts  
#HostbasedAuthentication no  
# Change to yes if you don't trust ~/.ssh/known_hosts for  
# HostbasedAuthentication  
#IgnoreUserKnownHosts no  
# Don't read the user's ~/.rhosts and ~/.shosts files  
#IgnoreRhosts yes  
  
# To disable tunneled clear text passwords, change to no here!  
#PasswordAuthentication yes  
#PermitEmptyPasswords no  
PasswordAuthentication yes <-- change yes to no (Yin Thu)  
  
# Change to no to disable s/key passwords  
#ChallengeResponseAuthentication yes  
ChallengeResponseAuthentication no  
  
# Kerberos options  
#KerberosAuthentication no  
#KerberosOrLocalPasswd yes  
#KerberosTicketCleanup yes  
-- INSERT --
```

(ssh အပိုင်း ၅) ssh ၏ default port no. အား ပြောင်းလဲ ခြင်း

ssh နဲ့ ပတ်သက်လို့ နောက်ဆုံး အပိုင်းပါ။ ကဲ ! ခပ်မြန်မြန် သွားရအောင်။ ပြောချင်တာက --> ISP တွေ Telecom တွေမှာ Data Center က server တွေဆီ operation center ကနေ ssh နဲ့ remote login ဝင်ပြီး configure လုပ်လေ့ ရှိကြတယ်။ ssh client ဘက်ကနေ server ဆီ ip ရယ် port no. ရယ်နဲ့ တွဲဝင်ရတယ်။

ssh ရဲ့ default port နဲ့ပါတ်က 22 ပဲ။

ပြောရရင် hacker တွေဆိုတာကလည်း အလစ်ကို ချောင်းနေကြတာ။ ssh ရဲ့ port ကို default အတိုင်း 22 လို့ ထားထားရင် hacker တွေ အတွက် penetrate လုပ်ဖို့ ခန့်မှန်းရ လွယ်သွားပြီ။ ဒါကြောင့် real world မှာ ssh ရဲ့ port ကို default အတိုင်း ထားလေ့ မရှိကြဘူး။ သောက်သုံးမကျတဲ့ ကောင်တွေဆိုရင်တော့ default အတိုင်း ထားရင်ထားမှာပေါ့လေ။ (ကြိုကား အဆဲတောင်းခြင်း) အဲတော့ hacker တွေ ခန့်မှန်းရ ခက်အောင် ဘယ်လို့ လုပ်မလဲ? မိမိ linux server ရဲ့ ssh port ကို တွေ့ခြား တစ်ခုခု ပြောင်းထားမယ်။ port no. တွေက 65,535 အထိ ရှိတယ်။ အဲအထဲက အဆင် ပြော တစ်ခု ရွှေးပြီး ပြောင်းထားရမယ်။ ဘယ်လို့ ပြောင်းမလဲ?

မိမိ linux server ကို root account နဲ့ login ဝင်။

၁။ `vi /etc/ssh/sshd_config`

၂။ **i key** ကို နှိပ်ပြီး **edit/insert mode** ထဲ ဝင်

၃။ **#Port 22** ဆိုတာကို ရှာ

၄။ ပြီးရင် # ဆိုတဲ့ သက်တကို ဖြုတ်

၅။ **22** နေရာမှာ **50110** လို့ ပြောင်း (**50110** အစား မိမိ အဆင်ပြောလိုလည်း ထည့်နိုင်ပါသည်)

၆။ **Esc** ကိုနိုင်

၇။ :wq လို့ ရှိက်သိမ်းပြီး ထွက် ဒါဆို ssh ရဲ့ default port no. ကို ပြောင်းလဲ ခြင်း process ပြီးဆုံးပြီ။ ထုံးစံအတိုင်း ပြောင်းထားတာလေး အသက်ဝင်သွားအောင် -->

`systemctl restart sshd` သို့မဟုတ် `systemctl reload sshd` အဲတော့ နောက်တခါ operation center ကနေ data center က server ဆီ ssh client နဲ့ ဝင်မယ်ဆို port no. နေရာမှာ 50110 လို့ ထည့်ပေးပြီး ဝင်ပေါ့။ ok !

အားလုံး အဆင်ပြောပါစေ။

Yin Thu

(15-6-2018)

```
# If you want to change the port on a SELinux system, you need to know about SELinux and how it handles port changes.
# If you want to change the port on a SELinux system, you need to know about SELinux and how it handles port changes.
# semanage port -a -t ssh_port_t -p tcp #PORT
#
#Port 22 ←
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
"/etc/ssh/sshd_config" 139L, 3905C
```

syslog

ကဲ ! ခုတစ်ခါ linux မှာ system နဲ့ဆိုင်တဲ့ log တွေအကြောင်း လေ့လာကြည့်ရအောင်။ linux မှာ log တွေကို ဖတ်တတ်ဖို့ အရေးကြီးပါတယ်။ မိမိ ဒဲ service တွေဟာ ပုံမှန် လည်ပတ်နေသလား? server တွေဆီ ဘယ်နေ့ ဘယ်အချိန်က ဘယ် user account တွေ login ဝင်သွားသလဲ? စသည်ဖြင့် log မှတ်တမ်းတွေမှာ ဝင်ကြည့်ရပါတယ်။ ဒါအပြင် log တွေဟာ problem တွေကို ဖြေရှင်းရာမှာ အသုံးဝင်ပါတယ်။ ကိုယ် run လိုက်တဲ့ service ဟာ ဘယ်နေရာမှာ error တက်သွားလဲ? error message က ဘာလဲ? solution ဘယ်လိုရှာမလဲ? ဒီ error code ကို ယူပြီး onlie မှာ solution ရှာမယ်။ စသည်ဖြင့်ပေါ့နော်။ log file တွေကို ဖတ်ရာမှာ root account နဲ့ ဝင်ဖတ်ရပါတယ်။

log ပိုင်းမှာ **rsyslog** နဲ့ **systemd-journal** လို့ log အမျိုးအစား ၂ ခု ရှိပြီး ဒီနေရာမှာ **rsyslog** အကြောင်းကို တင်ဆက်သွားမှာပါ။

#rsyslog

rsyslog ဆိုတာက အတိုပြောရရင် system, service စတာတွေနဲ့ သက်ဆိုင်တဲ့ message တွေ၊ log မှတ်တမ်းတွေကို မှတ်သားပေးတဲ့ open-source software utility တစ်ခုပါပဲ။

#log_rotation

ဒီနေရာမှာ log rotation အကြောင်း နည်းနည်း ကြားဖြတ် ပြောပါရစေ။ log rotation ဆိုတာက log file တွေကို ဘယ်နေ့ရှုံးရှုံးခြင်း ဘယ်အထိထားရှိမယ်၊ ဘယ်အချိန်ရောက်ရင်တော့ log အဟောင်းတွေကို ဖျက်ပစ်မယ်၊ အဲဒီလို သတ်မှတ် ထားတာကို ပြောတာပါ။ **logrotate** ဆိုတဲ့ **utility** ကို အသုံးပြုပြီး စီမံရပါတယ်။

#rsyslog

ဆက်ရရင် rsyslog ဟာ အဲဒီ log rotation လုပ်ချိန်က လွှဲပြီး system နဲ့ ဆိုင်တဲ့ log တွေကို [/var/log](#) ဆိုတဲ့ directory အောက်မှာ မှတ်သားထားပါတယ်။ [/var/log](#)

အဲဒီထဲကို ဝင်ပြီး console, mail, message, secure, cron, .. စသဖြင့် log တွေကို **ကြည့်ရနိုင်**ပါတယ်။

ဥပမာ system အကြောင်း message နဲ့ ဆိုင်တဲ့ log ကို **ကြည့်မယ်** ဆိုပါစို့။ အသုံးပြုရမယ့် command က အောက်ပါအတိုင်းပါ။ **less /var/log/messages** log message တွေရဲ့ လမ်းကြောင်း နဲ့ rules တွေကိုတော့ [/etc/rsyslog.conf](#) မှာ **ကြည့်ရနိုင်ပါတယ်**။

#log_priority_code

0 ကနေ 7 ထိ log ရဲ့ priority code တွေကတော့ အောက်ပါအတိုင်းပါပဲ။

0 >> **emerg** [system is unusable]

1 >> **alert** [action must be taken immediately]

2 >> **crit** [critical condition]

3 >> **err** [non-critical error condition]

4 >> **warning** [warning condition]

5 >> **notice** [normal but significant event]

6 >> **info** [informational event]

7 >> **debug** [debugging-level message]

#how_to_read_log

log တစ်ကြောင်းမှာ အပိုင်း င့် ခု ပါရှိပါတယ်။ log file တွေကို ဖတ်တဲ့အခါ အဲဒီ အပိုင်း င့် ခုကို မူတည်ပြီး ဖတ်ရမှာပါ။ အဲဒါတွေကတော့

1. log စတင် မှတ်သားတဲ့အချင်း

2. log message နဲ့ဆိုင်တဲ့ host ရဲ့ name

3. log message ကို ပေးပို့လိုက်တဲ့ program သို့မဟုတ် process

4. message ထဲမှာပါတဲ့ စာသား တို့ ဖြစ်ပါတယ်။

log file တွေကို **tail -f** အသုံးပြုပြီး live ကြည့်နိုင်ပါတယ်။

ဥပမာ security နဲ့ဆိုင်တဲ့ log ကို live ကြည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **tail -f /var/log/secure**

logger ဆိုတဲ့ command ကို အသုံးပြုပြီး log message တွေကို log file ဆီ ပေးပို့နိုင်ပါ သေးတယ်။

ဥပမာ user နဲ့ဆိုင်တဲ့ error message ကို ပေးပို့မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **logger -p user.err "this is testing"**

#customizing

log code တွေနဲ့ ဝင်ရောက်ရမယ့် လမ်းကြောင်းကို စိတ်ကြိုက် ပြောင်းလဲ သတ်မှတ်ပေးနိုင်ပါ သေးတယ်။

ဥပမာ error code တွေနဲ့ သူ့ရဲ့ လမ်းကြောင်းကို သတ်မှတ်ပေးမယ် ဆိုပါစို့။ error နဲ့ဆိုင်တဲ့ message တွေကို **/var/log/new-err** ဆိုတဲ့ file ထဲမှာ သိမ်းစေချင်တယ် ဆိုပါစို့။

ပြုလုပ်ရမယ့် အဆင့်တွေကတော့ အောက်ပါအတိုင်းပါ။

၁။ [/etc/rsyslog.d/](#) အောက်မှာ **new-err.conf** ဆိုတဲ့ file လေး ဆောက်ပါ။

၂။ **new-err.conf** ထဲမှာ ***.err /var/log/new-err** လို့ ရေးသား သတ်မှတ်ပေးလိုက်ပါ။

ဒီအချက် ၂ ချက်ကို command တစ်ကြောင်းတည်းနဲ့ ရေးနိုင်ပါသေးတယ်။ အောက်ပါအတိုင်းပေါ့။

echo "*.*.err /var/log/new-err" > /etc/rsyslog.d/new-err.conf

ကဲ ! ဟုတ်ပါပြီ။ အထက်က အဆင့်တွေအတိုင်း ပြီးသွားရင် rsyslog ကို restart ချပေးလုပ်မယ်။

`systemctl restart rsyslog`

ဒါဆို error နဲ့ သက်ဆိုင်တဲ့ message တွေဟာ `/var/log/new-err` ဆိုတဲ့ file ထဲ ရောက်သွားပါလိမ့်မယ်။

#testing_log_customization

အထက်ကအတိုင်း log location ကို စိတ်ကြိုက် သတ်မှတ်ပြီးရင်၊ မှန်ကန်မှ ရှိမရှိ `logger -p user.err "testing error log"` လို့ log message ပေးပို့ပြီးစမ်းကြည့်နိုင်ပါတယ်။

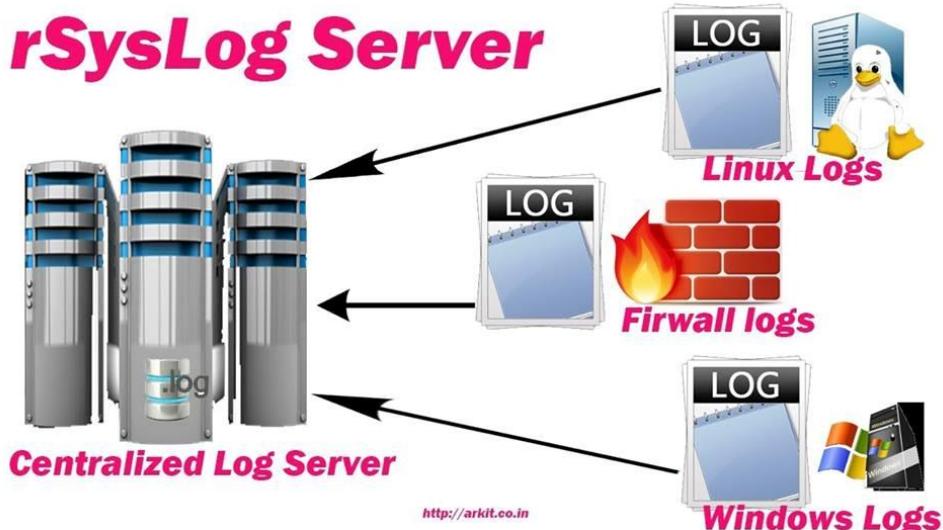
`logger -p user.err "testing error log"` error code ပို့ပြီးရင် `cat /var/log/new-err` လို့ ရှိက်ပြီးရောက်မရောက် ပြန်စစ်ရုံပေါ့။ `cat /var/log/new-err`

ကဲ ! rsyslog နဲ့ သက်ဆိုင်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(19-6-2018)



systemd-journal

linux system တစ်ခုမှာ log file တွေနဲ့ ပတ်သက်လို့ rsyslog နဲ့ systemd-journal ဆိုပြီး ဂုံးရှိပါတယ်။ rsyslog အကြောင်း ရေးတင်ပြီးပြီမ့် အခု ပြောမှာကတော့ systemd-journal အကြောင်းပါ။ systemd-journal ဟာ log တွေကို `/run/log` ထဲမှာ မှတ်သားပါတယ်။ system boot လုပ်ချိန်ကနေ စပြီး log တွေကို မှတ်သားထားပြီး system reboot လုပ်တာနဲ့ log တွေကို ဖျက်ပစ်ပါတယ်။ ဒါဟာ systemd-journal ရဲ့ သဘောသဘာဝပါပဲ။

၁။ systemd-journal log တွေကို `journalctl` ဆိုတဲ့ command ကို အသုံးပြုပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ `journalctl`

၂။ log တွေကို အရေအတွက်နဲ့ သတ်မှတ်ပြီး `journalctl -n` ဆိုတဲ့ option ကို တွဲသုံးရပါတယ်။ ဥပမာ ၅ ကြောင်းပဲ `journalctl -n` >> `journalctl -n 5`

၃။ log တွက် priority အလိုက် ခွဲခြားပြီး ကြည့်ဖို့ -p ဆိုတဲ့ option နဲ့ တွဲသံးရပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ journalctl -p err

ဒီနေရာမှာ သိထားဖို့က 0 ကနေ 7 ထိ log ရဲ့ priority code တွေ ရှိပါတယ်။ အဲဒါတွေကတေသာ အောက်ပါအတိုင်းပါပဲ။

0 >> **emerg** [system is unusable]

1 >> **alert** [action must be taken immediately]

2 >> **crit** [critical condition]

3 >> **err** [non-critical error condition]

4 >> **warning** [warning condition]

5 >> **notice** [normal but significant event]

6 >> **info** [informational event]

7 >> **debug** [debugging-level message]

၄။ log တွက် **live** အနေနဲ့ ဖတ်ချင်ရင် -f ဆိုတဲ့ option ကို တွဲသံးရပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါ။ journalctl -f

၅။ အချိန်ကန္တသတ်ချက်တစ်ခုအတွင်းက log တွက် ဖတ်ချင်ရင် --since နဲ့ --until ဆိုတဲ့ options တွက် သုံးစွဲရပါတယ်။ format ကတေသာ အောက်ပါအတိုင်းပါ။

--since Y-M-D H:M:S --until Y-N-D H:M:S

ဥပမာ 2017-02-19 05:00:00 အချိန်ကနေ 2017-02-19 06:00:00 အချိန်ကြားက log တွက် သိချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။

journalctl --since 2017-02-19 05:00:00 --until 2017-02-19 06:00:00

၆။ log တစ်ကြောင်းချင်းကို အသေးစိတ် **verbose mode** နဲ့ ကြည့်ချင်ရင် -o ဆိုတဲ့ option ကို သုံးရပါတယ်။ command syntax ကတေသာ အောက်ပါအတိုင်းပါပဲ။ journalctl -o

#saving_journalctl

log တွက် reboot လုပ်တိုင်း ဖျက်မပြစ်စေချင်ဘူး။ file တစ်ခုဆီ redirect လုပ်ထားမယ်ဆိုလည်း အဆင်ပြုပါတယ်။ ပြုလုပ်ရမယ့် အဆင့်တွေကတေသာ အောက်ပါအတိုင်းပါ။

၁။ </var/log/> အောက်မှာ **journal-log** အမည်နဲ့ directory တစ်ခု ဆောက်ပေးပါ။

၂။ chown :systemd-journal journal-log လို့ ရှိက်ပြီး **journal-log** ကို **systend-journal** ဆိုတဲ့ **group** က ပိုင်ဆိုင်ကြာင်း သတ်မှတ်ပေးလိုက်ပါ။

၃။ chmod g+s journal-log လို့ ရှိက်ပြီး special permission သတ်မှတ်ပေးလိုက်ပါ။

၄။ killall _USR1 systemd-journald လို့ ရှိက်ပြီး restart လုပ်ပေးလိုက်ပါ။

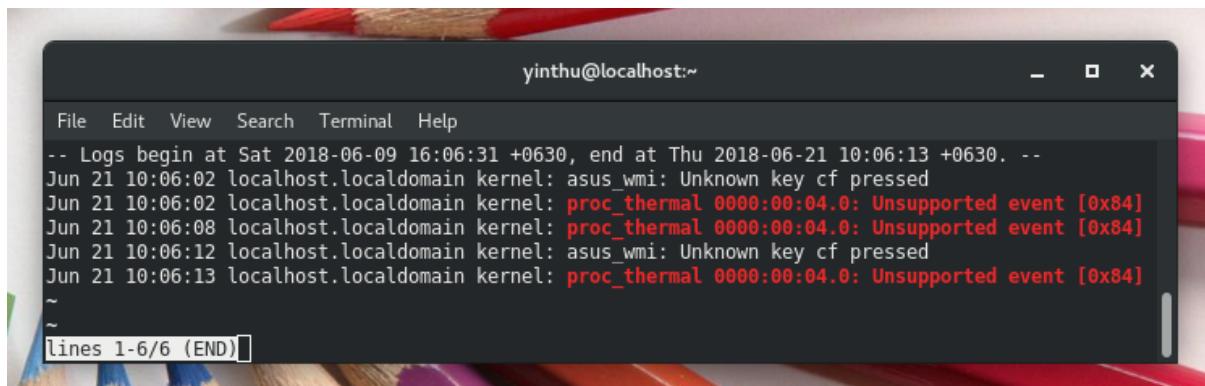
ဒါလို log တွေကို reboot လုပ်တာနဲ့ ဖျက်မပစ်ပဲ </var/log/> အောက်က journal-log ဆိုတဲ့ directory ထဲမှာ မှတ်သားပေးသွားပါလိမ့်မယ်။

ကဲ ! systemd-journal အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(21-6-2018)



```
yinthu@localhost:~  
File Edit View Terminal Help  
-- Logs begin at Sat 2018-06-09 16:06:31 +0630, end at Thu 2018-06-21 10:06:13 +0630. --  
Jun 21 10:06:02 localhost.localdomain kernel: asus_wmi: Unknown key cf pressed  
Jun 21 10:06:02 localhost.localdomain kernel: proc_thermal 0000:00:04.0: Unsupported event [0x84]  
Jun 21 10:06:08 localhost.localdomain kernel: proc_thermal 0000:00:04.0: Unsupported event [0x84]  
Jun 21 10:06:12 localhost.localdomain kernel: asus_wmi: Unknown key cf pressed  
Jun 21 10:06:13 localhost.localdomain kernel: proc_thermal 0000:00:04.0: Unsupported event [0x84]  
~  
~  
lines 1-6/6 (END)
```

🍁 The True Happiness is Playing with Linux

linux system administration ပိုင်းကို လေ့လာမယ်ဆိုရင် GUI (graphical user interface) မသုံးမယ့်တူတူ centos linux ကို minimal နဲ့ တင်တာ ပိုကောင်းတယ်။ CLI (command line interface) ပေါ့။ RAM လည်း စားတာ အရမ်း သက်သာမယ်။ ပိုလည်း မြန်မယ်လေ။ အဓိက က linux ဆိုတာ တင်ပြီး run ထားရှိနဲ့ ကိစ္စ ပြီးတာ မဟုတ်ဘူး။ os version အလိုက် package installation procedure/server configuration method တွေက ပြောင်းနေတာ။ virtual box မှာ စမ်းနေရင်း တန်းလန်း online လေးပါ တက်ပြီး how to install ဘာညာ၊ how to configure ဘာညာ၊ ရှာရဖွေရည်းမှာ။ စက်လေးနေရင် အဆင်မပြောဘူး။စက်ကို RAM မြှင့်ဖို့ကလည်း ပိုက်ဆံက မရှိဘူး။ဒီတော့ ပေါ့ပါးမယ့် နည်းလမ်း သုံးဖို့လိုတယ်လေ။

Admin က fedora linux သုံးတယ်။ virtual box ထပ်သွင်းပြီး centos vm ၂ ခု အောက်ထားတယ်။ centos 7-1 နဲ့ centos 7-2 ဆိုပြီး ၂ ခုလုံး minimal နဲ့ သွင်းထားတယ်။ အပြင်က host os နဲ့ vbox ပေါ်က os ၂ ခုကို virtual network ချိတ်ထားတယ်။ စက် ၃ လုံး connection မိနေတဲ့ သဘောပေါ့။ ဘယ်လို သုံးလဲဆိုတော့ -->

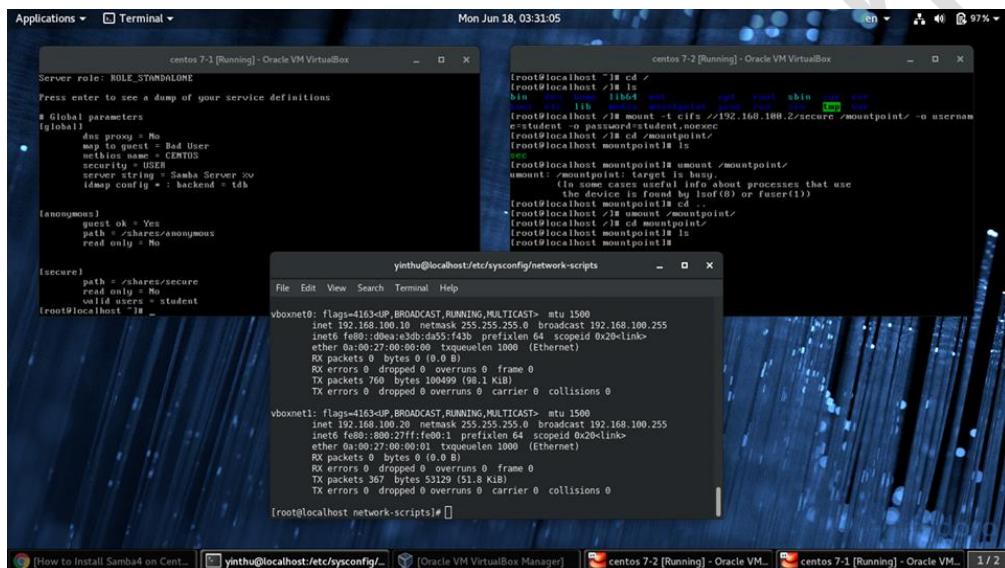
၁။ nfs server, samba server, web, dhcp စသည်ဖြင့် စမ်းမယ်ဆို centos 7-1 vm မှာ တင်ပြီး စမ်းတယ်။ သူက server ပေါ့။ centos 7-2 vm က client ပေါ့။

၂။ os level administration စမ်းမယ်ဆို centos 7-2 vm တစ်ခုတည်းကို သုံးတယ်။

ဆိုလိုချင်တာက ကိုယ့်စက်မှာ RAM နည်းနေတဲ့ အချိန် virtual box or vmware မှာ linux ကို GUI နဲ့ vm ဂုံးသွင်းမယ်။ ဂုံးလုံး run မယ်ဆို စက်လေးသွားမှုပါ။ အဲဒီအစား၊ minimal installation နဲ့ သွင်းပြီး စမ်းရတာဟာ အတော် အဆင်ပြေပါတယ်။ GUI ထက်လည်း အဆများစွာ ပိုမြန်ပါတယ်။ စာတွေ စမ်းဖို့ အလုပ်နဲ့ဆိုင်တဲ့ POC တွေ စမ်းဖို့ အတွက် အသုံးဝင်ပါတယ်။ Admin ရဲ့နည်းလမ်းကို မျှဝေရုပါ။ ပိုကောင်းတာတွေလည်း ရှိရင် ရှိမှာပေါ့။ ဒီလောက်ပါပဲ။ linux ကို လေ့လာ စမ်းသပ် ကလိုရင်း ပျော်စရာတွေ ဖန်တီးနိုင်ပါစေ လို့။

Yin Thu

(21-6-2018)



#####network configuration with nmcli

linux မှာ network setting ကို configure လုပ်တဲ့အကြောင်း လေ့လာကြရအောင်။ network card မှာ သတ်မှတ်ပေးထားတဲ့ ip, mac address စတာတွေကို **ifconfig** ဆိုတဲ့ command နဲ့ ခေါ်ကြည့် နိုင်ပါတယ်။ **ifconfig** မရှိရင် **yum -y install net-tools** နဲ့ သွင်းယူပါ။

#how_to_configure

network setting တွေကို သတ်မှတ်ပေးဖို့ အတွက်ကတော့ နည်းလမ်း ဂုံးရှိပါတယ်။

၁။ တစ်ခုက **nmcli** (network manager command line interface) ဆိုတဲ့ command ကို အသုံးပြုပြီး command line ကနေ တစ်ဆင့်ချင်း သတ်မှတ်သွားတဲ့ နည်းလမ်းပါ။

၂။ နောက်တစ်ခုက network file ထဲကို **vim** ဆိုတဲ့ command line text editor နဲ့ manually ဝင်ပြင်တဲ့ နည်းလမ်းပါ။

ဒီနေရာမှာ အရင်ဆုံး **nmcli** အကြောင်းကို တင်ဆက်ပါမယ်။

#nmcli

၁။ **nmcli** ကို အသုံးပြုပြီး configure လုပ်မယ်ဆိုရင် **terminal** ထဲမှာ **nmcli** လို ရှိက်ပြီး **tab key** ကို ပျက် ဆက်တိုက် နှုပ်ပေးလိုက်ပါ။ **nmcli** နဲ့ ဆက်စပ် သုံးစွဲမယ့် **options** တွေကို ပြပေးပါ လိမ့်မယ်။

၂။ ကျေလာတဲ့ **options** တွေထဲကမှ **connection** ကို **configure** လုပ်မှာဆို **connection** ကို ရွှေး
tab key ပျက် ထပ်နိပ်၊ **connection** နဲ့ ဆက်နှယ်တဲ့ **add modify show** စသဖြင့် **options** တွေ ထပ်ကျေလာပါလိမ့်မယ်။

၃။ **connection** အသစ် ဆောက်မှာဆို **add** ကို ရွှေးပြီး **tab key** ပျက် ထပ်နိပ်၊ **setting** တွေ ဖြည့်။

၄။ ရှိပြီးသား **connection** ကို ပြုပြင်မှာဆို **modify** ကို ရွှေးပြီး **tab key** ပျက် ထပ်နိပ်၊ **setting** တွေ ဖြည့်။

၅။ **connection** အခြေအနေကို ကြည့်မှာဆို **show** ကို ရွှေး **enter** ခေါက်။

ဒီလို တစ်ဆင့်ချင်း သွားရပါတယ်။ **nmcli** နဲ့ ဆိုင်တဲ့ **options** တွေကို အလွတ်ကျက်နေစရာ မလိုအပ်ပါဘူး။ ဒါက **nmcli** ကို သုံးပြီး **network** ကို **configure** လုပ်တဲ့ သဘောတရားပါပဲ။

#example

ဥပမာ အနေနဲ့ **nmcli** ကို သုံးပြီး **connection** အသစ် ၁ ခု create လုပ်ကြည့်ကြစိုး။

```
IP >> 192.168.100.10  
NETMASK >> 255.255.255.0  
GATEWAY >> 192.168.100.254  
DNS >> 8.8.8.8
```

အထက်ပါ အချက်တွေ အတိုင်း **connection** တစ်ခု create လုပ်ကြည့်ကြမယ်နော်။ လွယ်လွယ် လေးပါ။ command တစ်ကြောင်းတည်းနဲ့ ကရှုစပြီးပါတယ်နော်။ တစ်ခုခု အစပျိုးပြီးတိုင်း Tab key လေးကို ခုန်ကြည့်စေခဲ့တယ်။ ပိုလွယ်တာပေါ့။

```
nmcli connection add ifname test type Ethernet ipv4.addresses 192.168.100.10/24  
ipv4.gateway 192.168.100.254 ipv4.dns 8.8.8.8
```

ပြီးရင် **enter** ခေါက်ရုံပေါ့။ မိမိ ရေးသွင်းတာတွေက မှန်တယ် ဆိုရင် **Connection 'ethernet-test'** **(d64a1a32-6fdd-47ee-9f15-ed891fffdbe1f)** successfully added.

လို ပြပါလိမ့်မယ်။

မိမိ တည်ဆောက်လိုက်တဲ့ **connection** ကို **nmcli connection show** ဆိုတဲ့ command နဲ့ ပြန်စစ် နိုင်ပါသေးတယ်။

```
nmcli connection show
```

မိမိ တည်ဆောက်လိုက်တဲ့ **connection** ကို တစ်ခုခု ထပ်မံ ပြင်ဆင်မယ် (ဥပမာ ip ပြောင်းမယ်) ဆို **nmcli connection modify** လို့ ရိုက် **Tab key** ၂ ချက် ခုနှုန်း ကျလာတာတွေထဲက **modify** လုပ်မယ့် **connection name** ကို ထည့်ပေးပြီး ပြင်စရာရှိတာ ပြင်ပေါ့။

ပြန်ဖျက်မယ် ဆိုရင်တော့ **nmcli connection delete connection-name** ပေါ့။ ဒီနေရာမှာ **connection-name** က **ethernet-test** ဆိုတော့ ဖျက်မယ်ဆို အောက်ပါအတိုင်းပေါ့။

nmcli connection delete ethernet-test

#path

မိမိ တည်ဆောက်လိုက်တဲ့ connection က ဘယ်နေရာမှာလဲ? ဆိုတော့ >>

/etc/sysconfig/network-scripts/ အောက်မှာ **ifcfg-ethernet-test** ဆိုတဲ့ **connection** အသစ် ၁ ခု အဖြစ် ရှိနေပါလိမ့်မယ်။ အဲထဲ ဘာတွေ ရှိလဲ သိချင် **vi** နဲ့ဖွင့်ကြည့်ပေါ့နော်။

vi /etc/sysconfig/network-scripts/ifcfg-ethernet-test

ကဲ ! nmcli ကို အသုံးပြုပြီး connection တစ်ခု တည်ဆောက်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။ တွဲတင်ထားတဲ့ photo ကို zoom နဲ့ကြည့်မယ်ဆို ပိုသိ ပိုလွှယ်ကူးသွားပါလိမ့်မယ်နော်။ nmcli အကြောင်း ပိုပြီး သိချင်ရင် အောက်ပါ link မှာ သွားရောက် လေ့လာနိုင်ပါတယ်။

<https://www.tecmint.com/configure-network-connections-using-nmcli-tool-in-linux/>

အားလုံး အဆင်ပြေပါစော်။

Yin Thu

(22-6-2018)

```
yinthu@localhost ~]$ nmcli connection add ifname test type Ethernet ipv4.addresses 192.168.100.10/24 ipv4.gateway 192.168.100.254 ipv4.dns 8.8.8.8
Connection 'ethernet-test' (d64ala32-6fdd-47ee-9f15-ed891ffdb1f) successfully added.
[yinthu@localhost ~]$ nmcli connection
add delete edit help load monitor show
clone down export import modify reload up
[yinthu@localhost ~]$ nmcli connection show
NAME           UUID                                  TYPE      DEVICE
Wired connection 2 5ddcd24d9-3d46-3074-aa7e-13fa4f3b612e  ethernet  enp0s20u3
PC-Faster-1411 38835656-3292-4796-9fda-010f935660f2  wifi     --
Wired connection 1 bb9044e9-e423-465f-8ef5-32ef425dff89  ethernet  --
ethernet-test   d64ala32-6fdd-47ee-9f15-ed891ffdb1f  ethernet  --
local NIC       8c6fd7b1-ab62-a383-5b96-46e083e04bb1  ethernet  --
```

network configuration (manually)

nmcli ကို အသုံးချပြီး network configuration အကြောင်းပြီးတဲ့နောက်၊ ခုတခါ network ကို **vi** နဲ့ manually ဝင်ပြင်တဲ့အကြောင်းပြုရအောင်။

#concept

network config ကို **vi** နဲ့ ဝင်ပြင်တဲ့ နည်းကတော့ ရှိုးရှင်းပါတယ်။ **network file** တွေဟာ </etc/sysconfig/network-scripts> ဆိုတဲ့ **directory** အောက်မှာ တည်ရှိပါတယ်။

cd /etc/sysconfig/network-scripts ls

connection အသစ် ၁ ခု ဆောက်ချင်ရင်၊ ပေးချင်တဲ့ အမည်ရှေ့မှာ **ifcfg-** ခံပြီး **touch** နဲ့ တည်ဆောက်ပေးရပါတယ်။ **ifcfg** နဲ့စတဲ့ **file** တွေဟာ network config file တွေလို့ နားလည်ပါမယ်။ ဥပမာ **eth0** ဆိုတဲ့ အမည်နဲ့ **connection** အသစ် ၁ ခု ဆောက်ချင်တယ် ဆိုပါစဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **touch ifcfg-eth0**

ရှိပြီးသား **connection** ကို ပြင်ဆင်ချင်ရင်တော့ network file (ဥပမာ **ifcfg-eth0** ကို **vi** နဲ့ဖွင့်ပြီး ဝင်ရောက် ပြင်ဆင်နိုင်ပါတယ်။ **vi ifcfg-eth0**

အဲဒီအခါ **ifcfg-eth0** ထဲမှာ network setting တွေ၊ TYPE, DEVICE, NAME, HWADDR, ONBOOT, BOOTPROTO, IPADDR, NETMASK, GATEWAY, DNS1, DNS2 စတာတွေရဲ့ တန်ဖိုးတွေကို ထည့်သွင်းပေးရပါမယ်။

ဒါကတော့ **nmcli** မသုံးပဲ network file တွေကို manually ဖန်တီး၊ **vi** နဲ့ ဝင်ပြင်တဲ့ သဘောတရားပါပဲ။

#example

```
IP >> 192.168.100.10  
NETMASK >> 255.255.255.0  
GATEWAY >> 192.168.100.254  
DNS1 >> 8.8.8.8  
DNS2 >> 8.8.4.4
```

အထက်ပါ ပေးထားချက်တွေအတိုင်း network connection တစ်ခုကို **vi** နဲ့ create လုပ်ရအောင်။ ၁။ network config file တွေ ရှိတဲ့ </etc/sysconfig/network-scripts> ဆိုကို သွားပါ။

cd /etc/sysconfig/network-scripts ls

j။ **ifcfg-eth0** (new version မှာ **ifcfg-enp0s2** or **ifcfg-enp0s3**) ဆိုတဲ့ **file** ကို **vi** နဲ့ဖွင့်ပါ။
vi ifcfg-eth0

၃။ ပြီးရင် အောက်ပါအတိုင်း အစဉ်လိုက် ထည့်ပေးပါနော်။

TYPE=Ethernet (connection အမျိုးအစား)

DEVICE=eth0 (device အမည်)

NAME="local NIC" (မိမိ နှစ်သက်သလို ပေးနိုင်သည်)

HWADDR=fe80::800:27ff:fe00:0 (hardware id or mac address)

BOOTPROTO=static (ip ကို မိမိကိုယ်တိုင် သတ်မှတ်မှုဆို ဒီနေရာမှာ static လို့ ထည့်။ dhcp နဲ့ auto သတ်မှတ်မှုဆို dhcp လို့ ထည့်။)

ONBOOT=yes (စက်ဖွင့်ဖွင့်ချင်း အလုပ်လုပ်ဖို့)

NM_CONTROLLED=yes (GUI network manager ကနေ လှမ်းပြင်နိုင်စေချင်ရင် yes ..မပြင်စေချင်ရင် no)

IPADDR=192.168.100.10 (မိမိ ပေးချင်တဲ့ ip)

NETMASK=255.255.255.0

GATEWAY=192.168.100.254 (gateway ip .. လိုရင်ထည့် မလို မထည့်နဲ့ပေါ့။)

DNS1=8.8.8.8 (domain name server 1 ရဲ့ ip)

DNS2=8.8.4.4 (domain name server 2 ရဲ့ ip)

ငါ၏ ဒါတွေ ဖြည့်ပြီးပြီဆိုရင် Esc key နှင့် :wq ရိုက် enter ခေါက်ပြီး ထွက်ရုံပေါ့။

ဒါဆို မိမိ လိုအပ်ချက်တွေနဲ့ အညီ connection ၁ ခု ကို manually တည်ဆောက်လို့ ပြီးပါပြီ။

#remark

network ကို ပြင်ဖို့ root account သို့မဟုတ် administrator account ကို သုံးဖို့ မမေ့ပါနဲ့။

connection အသစ်တည်ဆောက်တာဖြစ်စေ ရှိပြီးသားကို ပြုပြင်တာဖြစ်စေ၊ ကိစ္စပြီးသွားရင်

configuration တွေ အသက်ဝင်စေဖို့ network service ကို restart ချပေးဖို့မမေ့ပါနဲ့။

systemctl restart network သို့မဟုတ် service network restart

network setting ရဲ့ IPADDR နေရာမှာ ထည့်ထားတဲ့ IP မှန်မမှန် သိဖို့ ping ကို သုံးပြီး စစ်ဆေး နိုင်ပါတယ်။ ping 192.168.100.10

network setting ရဲ့ DNS1 နေရာမှာ ထည့်ထားတဲ့ dns ရဲ့ ip ဝင်မဝင်၊ /etc/resolv.conf ကို ဝင်ရောက် ကြည့်ရှုနိုင်ပါတယ်။ cat /etc/resolv.conf ပေါ့။

ကဲ ! network config ကို vi နဲ့ manually ဝင်ရေးတဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(22-6-2018)

```
File Edit View Search Terminal Help
yinthu@localhost:~>
TYPE=Ethernet
DEVICE=enp2s0
NAME="local NIC"
HWADDR=1C:87:2C:94:E3:17
BOOTPROTO=static
ONBOOT=yes
NM_CONTROLLED=yes

IPADDR=192.168.100.1
NETMASK=255.255.255.0
GATEWAY=192.168.100.254

DNS1=8.8.8.8
DNS2=8.8.4.4
```

hostname

မိမိစက်ရဲ ip ကို network ပေါ်မှာ host name နဲ့ ping လို့ရအောင် ဖန်တီးနိုင်ပါတယ်။
host name ဆိုတာက ip အစားသုံးနိုင်တဲ့ စက်ရဲ အမည်ပေါ့။
နည်းလမ်းလေးတွေကို ကြည့်ရ အောင်။

#method_1

/etc/hosts ဆိုတဲ့ file မှာ ဝင်ရောက် ပြင်ဆင်ရုပါ။

ဥပမာ ip က 192.168.100.10 ၊ ဒါကို www.example.com လို့ ping ရင်လည်း reply ပြန်စေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။
echo "192.168.1.254 www.example.com" >> /etc/hosts

ဒါဆို ping www.example.com လို့ ping ရင်လည်း reply ပြန်ပါပြီ။

#method_2

ဒါအပြင် linux machine ရဲ hostname ကို ပြောင်းလဲမယ်ဆိုရင် hostnamectl set-hostname ဆိုတဲ့ command ကို အသုံးပြုနိုင်ပါတယ်။

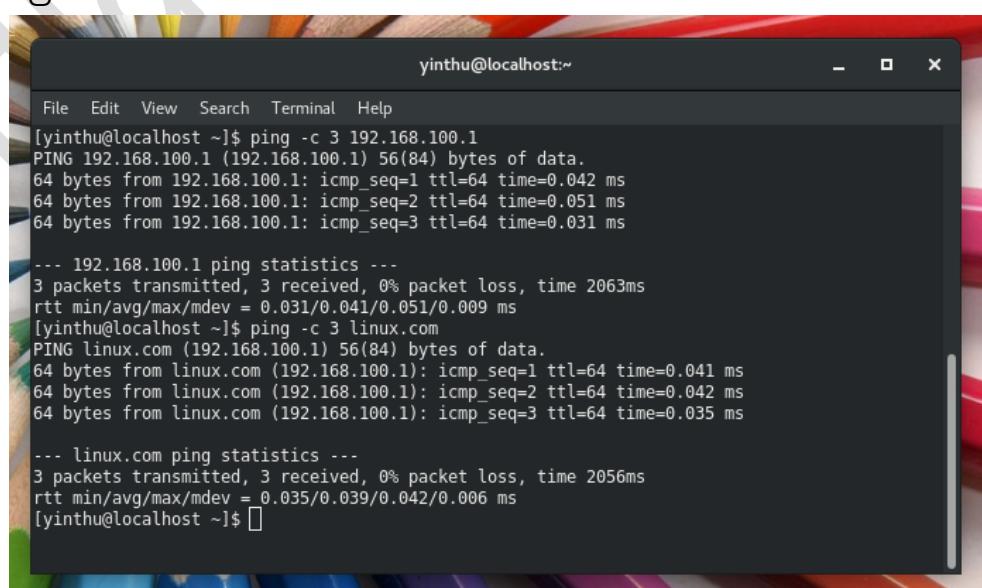
ဥပမာ hostname ကို test.com လို့ ပြောင်းလဲမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ hostnamectl set-hostname test.com

#method_3

vim command line text editor နဲ့ /etc/hostname ဆိုတဲ့ file ဖွင့်ပြီး ဝင်ရောက် ပြောင်းလဲပေး နိုင်ပါတယ်။ ဥပမာ hostname ကို linux.com လို့ ပြောင်းလဲမယ် ဆိုပါစို့။ vi /etc/hostname
ကို နိုပ် insert mode ပြောင်း linux.com လို့ ရေး၊ Esc key ကို နိုပ်၊ :wq နဲ့ save and exit လုပ်ပြီး ထွက်။ ကဲ ! ip အစား hostname သတ်မှတ်ပေးတဲ့ အပိုင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(22-6-2018)



```

yinthu@localhost:~$ ping -c 3 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.031 ms

--- 192.168.100.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2063ms
rtt min/avg/max/mdev = 0.031/0.041/0.051/0.009 ms
[yinthu@localhost ~]$ ping -c 3 linux.com
PING linux.com (192.168.100.1) 56(84) bytes of data.
64 bytes from linux.com (192.168.100.1): icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from linux.com (192.168.100.1): icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from linux.com (192.168.100.1): icmp_seq=3 ttl=64 time=0.035 ms

--- linux.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.035/0.039/0.042/0.006 ms
[yinthu@localhost ~]$

```

root password recovery (redhat, centos, fedora)

linux မှာ root account ရဲpassword ဟာ အရေးကြီးပါတယ်။ အဲဒီ root password ကို မေ့သွားပြီဆိုပါတော့။ ဘယ်လို လုပ်ကြမလဲ? ဒီနေရာမှာ root password recovery ဟာ အရေးပါလာပါတယ်။ ကဲ ! စလိုက်ကြရအောင်။ root account ရဲpassword ကို မေ့သွားပြီပဲ ထားပါ။

လုပ်ရမယ့် အဆင့်တွေက -->

၁။ system ကို reboot ချပါ။

၂။ ပြန်တက်လာလို့boot menu ပေါ်လာရင် ကြိုက်တဲ့ key တစ်ခုကို နှိပ်ပြီး brake ဖမ်းလိုက်ပါ။

၃။ keyboard ပေါ်က e ကို နှိပ်ပါ။

၄။ systemctl reboot လို့ ရှိက် enter ခေါက်ပါ။ password အသစ်နဲ့ boot ပြန်တက်လာပါလိမ့်မယ်။ စာပိုဒ်ရဲ့အဆုံးကိုသွားပါ။

၅။ **console=tty1 rd.break** လို့ ရှိက်ပါ။

၆။ **ctrl+x** ကို နှိပ်ပြီး ရွှေ့ဆက်ပါ။

၇။ root အနေနဲ့ login ဝင်သွားပါလိမ့်မယ်။

၈။ **mount -o remount,rw /sysroot/** လို့ ရှိက် enter ခေါက်ပါ။

၉။ **chroot /sysroot/** လို့ ရှိက် enter ခေါက်ပါ။

၁၀။ sh shell ထဲကို ဝင်သွားပါလိမ့်မယ်။

၁၁။ **passwd root** လို့ ရှိက် enter ခေါက်ပါ။

၁၂။ **password** အသစ် ထည့်ပေးလိုက်ပါ။

၁၃။ / (root directory) အောက်မှာ အောက်ပါအတိုင်း file ၁ ခု ဆောက်ပေးလိုက်ပါ။ **touch ./autorelabel**

၁၄။ **ပြီးရင် exit** လို့ ရှိက် enter ခေါက်ပြီး ထွက်ပါ။

၁၅။ **systemctl reboot** လို့ ရှိက် enter ခေါက်ပါ။

၁၆။ password အသစ်နဲ့ boot ပြန်တက်လာပါလိမ့်မယ်။ ဒါပါပဲ။ root account ရဲpassword မေ့သွားရင် ဒီနည်းလေးကို သတိရကြပါလို့။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(22-6-2018)

```

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/# mount -o remount,rw /sysroot
switch_root:/# chroot /sysroot
sh-4.2# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 7 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-4.2# touch ./autorelabel
sh-4.2# Have A Nice Time All My Facebook Friends (Yin Thu)
sh-4.2# exit

```

scp (secure copy)

ဒီတစ်ခါ ပြောမှာကတော့ copy ကူးတဲ့အကြောင်းပါပဲ။ ဒါပေမယ့် သူက ရှိုးရှိုး copy မဟုတ်ပါဘူး။ secure copy ပါ။ scp ပေါ့။ scp ကို network ချိတ်ထားတဲ့ linux system တစ်ခုနဲ့တစ်ခု အကြား file တွေကို ကူးယူရာမှာ လုံခြုံမှု ရှိစေဖို့ သုံးပါတယ်။ network ပေါက စက်အချင်းချင်း file အပို့/အယူ လုပ်ရာမှာ အချိန်ကုန် သက်သာစေပါတယ်။ ssh service ကို run ထားဖို့တော့ လိုတာ ပေါ့နော်။

ဥပမာ A ော် **tom** ဆိုတဲ့ user သည် network ပေါက တစ်ခြားစက်တစ်လုံးရဲ့ **jerry** ဆိုတဲ့ user ဆီက </home/jerry/> အောက်မှာရှိတဲ့ **file1** ကို လုမ်းကူးယူမယ်။ **jerry** ရဲ့ ip က 192.168.100.3 ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

```
scp jerry@192.168.100.3:/home/jerry/file1 /home/tom
```

jerry ရဲ့ password ကို တောင်းပါလိမ့်မယ်။ ထည့်ပေးလိုက်ပေါ့။

ဥပမာ B ော် **tom** ဆိုတဲ့ user သည် သူရဲ့ </home/tom/> အောက်က **file1** ကို network ပေါက **jerry** ဆိုတဲ့ user ရဲ့ </home/jerry> ဆီကို လုမ်းပိုမယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

```
scp /home/tom/file1 jerry@192.168.100.3:/home/jerry
```

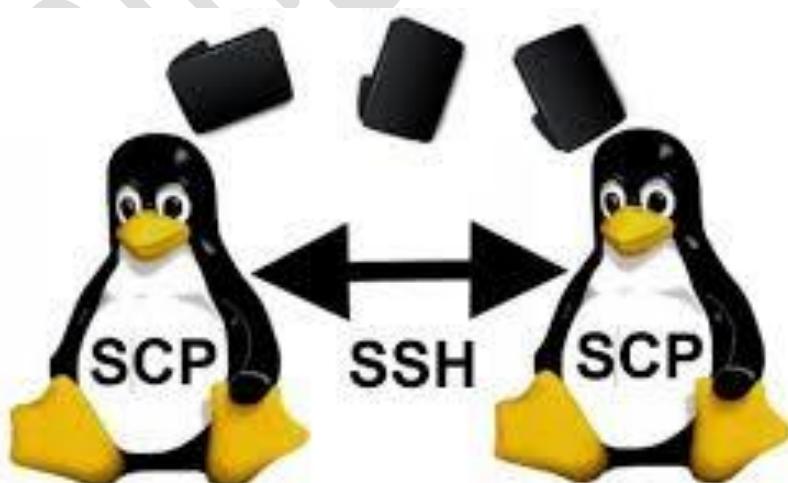
jerry ရဲ့ password တောင်းပါလိမ့်မယ်။ ထည့်ပေးလိုက်ပေါ့။ ကဲ ! ဟုတ်ပါပြီ။ ဆိုရရင်

အထက်မှာ ပြောတဲ့ ဥပမာ ၂ ခုအတိုင်းပေါ့။ scp ရဲ့ အလုပ်လုပ်ပုံကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(27-6-2018)



rsync ဖြင့် ကူးယူခြင်း

rsync ...။ သူက copy ကူးသလိုပါပဲ။ ဒါပေမယ့် ရှိုးရှိုး copy နဲ့ မတူတာက၊ သူက synchronize လုပ်ပြီး ကူးတာပါ။ ဒါကြောင့် သူကို rsync (remote synchronization) လို့ ခေါ်တာပါ။

ဥပမာ file 10 ခုကို memory stick ထဲ ကူးထည့်တယ်။ ရှုတ်တရက် ကူးရှင်းတန်းလန်း memory stick ပြုတ်သွားတယ် ဆိုပါစို့။ ဒီနေရာမှာ rsync ဟာ file တွေကို အစအဆုံး ပြန်ကူး၊ file ထပ်နေရင် replace လုပ်စရာ မလိုပဲ၊ လိုအပ်တဲ့ နေရာကနေ synchronize လုပ်ပြီး ကူးသွားပါလိမ့်မယ်။ အချိန်နည်းနည်း ပိုကြာပေမယ့် save ဖြစ်ပါတယ်။ ဒါကတော့ rsync ခဲ့ အလုပ်လုပ်တဲ့ သဘော တရားပါပဲ။ rsync ကို အသုံးပြုချင်တဲ့ အခါ သမားရှိုးကျေ cp ဆိုတဲ့ command အစား **rsync** ကို -av ဆိုတဲ့ option နဲ့ အသုံးပြုလိုက်ရုပါပဲ။

ဥပမာ **current directory** ထဲက saisai.mp3 ဆိုတဲ့ file ကို /mnt/ အောက်က songdrive ဆို **rsync** နဲ့ ကူးမယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

rsync -av snow.mp3 /mnt/songdrive

ဥပမာ **tom** ဆိုတဲ့ user သည် network ပေါ်က တစ်ခြားစက်တစ်လုံးရဲ့ **jerry** ဆိုတဲ့ user ဆိုက /home/jerry/ အောက်မှာရှိတဲ့ **file1** ကို လုမ်းကူးမယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **rsync jerry@host.com:/home/jerry/file1 /home/tom/**

jerry ရဲ့ password ကို တောင်းပါလိမ့်မယ်။ ထည့်ပေးလိုက်ပေါ့။ ယခင် သုံးနေကျေ **cp/scp** အစား **rsync -av** ကို အသုံးပြုသွားတဲ့ သဘောပါပဲနော်။ rsync အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(28-6-2018)



sftp (secure file transfer protocol)

ခုတစ်ခါ ပြောမှာကတော့ file transfer protocol အကြောင်းပါ။ ရှိုးရှိုး ftp မဟုတ်တဲ့ secure ဖြစ်တဲ့ sftp ပေါ့။ copy ကူးတာထက် ပိုအဆင်ပြောပါတယ်။ ဒါပေမယ့်လည်း အသုံးနည်းပါတယ်နော်။ filezilla တို့ဘာတို့ftp client တွေကိုပဲ အသုံးများပါတယ်။ ဒါပေမယ့် သိတားသင့်လို့ တင်ပေးလိုက်ပါတယ်နော်။

#installation

sftp ကို အသုံးပြုနိုင်ဖို့ **vsftpd** ကို install လုပ်ရပါတယ်။ မရှိခဲ့ရင်ပေါ့။ **yum install vsftpd**
ပြီးရင် **systemctl enable vsftpd** **systemctl start vsftpd**

#concept

network ပေါ်က စက်တွေ တစ်ခုနဲ့ တစ်ခု အကြား **sftp** အသုံးပြုပြီး **file** တွေကို **transfer** လုပ်ရာမှာ **file upload** | **file download** ပုံစံနွားပါတယ်။ **upload** လုပ်ချင်ရင် **put** ဆိုတဲ့ command ကို သုံးစွဲရပါတယ်။ **download** လုပ်ချင်ရင် **get** ဆိုတဲ့ command ကို သုံးစွဲရပါတယ်။

#examples

ဥပမာ **jerry** ရဲ့စက်လဲကို မိမိရဲ့ current directory အောက်က **file1** ဆိုတဲ့ **file** ကို ပိုမယ် ဆိုပါစို့။ **jerry** ရဲ့ ip က 192.168.100.1 ထားပါတော့။ **jerry** နဲ့ ချိတ်ဆက်မိဖို့ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **sftp jerry@192.168.100.1**

ဒါဆိုရင် sftp ရဲ့ custom shell လေး ကျေလာပါလိမ့်မယ်။ အောက်ပါပုံစံမျိုးနဲ့ပေါ့။ **sftp>**
အဲဒီ **shell** ထဲမှာ **ls**, **mkdir** စတဲ့ command တွေ အသုံးပြုနိုင်ပါတယ်။ အစပြန်ကောက်ရရင် **jerry** ဆီ **file upload** လုပ်မှာ ဖြစ်တဲ့အတွက် **put** ဆိုတဲ့ command ကို သုံးရပါမယ်။

sftp> put file1 ပေါ့။ ဒီအခါ **file1** ဟာ **jerry** ဆီ ရောက်သွားပါလိမ့်မယ်။ ဒါက **file** ပိုတဲ့ အပိုင်းပါ။
file ကို **download** ဆွဲချင်ရင် **get** ဆိုတဲ့ command ကို သုံးရပါတယ်။

ဥပမာ **jerry** ဆီက **file2** ဆိုတာလေးကို **download** ပြန်ဆွဲမယ် ဆိုပါစို့။ **download** ဆွဲမှာ ဖြစ်လို့ အသုံးပြုရမယ့် **command** က **get** ပါ။ **sftp> get file2** ပေါ့။

ဒါဆို **jerry** ဆီက **file2** ဟာ ကိုယ့်ရဲ့ **current directory** ထဲ ရောက်လာပါလိမ့်မယ်။

တို့တို့ရှင်းရှင်းပါပဲနော်။ **sftp** အသုံးပြုပြီး network ပေါ်မှာ file တွေ ပေးပို့ (put) ရယူ (get) တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြောပါစေ။

Yin Thu

(1-7-2018)



SFTP over a secure channel, such as SSH

identifying file system and device

ခုတခါ linux မှာ file system နဲ့ device တွေကို စစ်ဆေးတဲ့အကြောင်း လေ့လာကြည့်ရအောင်။ စာနဲ့ ရှုပ်သလို ထင်ရပေမယ့် သေချာလေး အေးအေးဆေးဆေး ဖတ်သွားပေးပါနော်။ linux မှာ တပ်ဆင်ထားတဲ့ hard disk, cd drive စတဲ့ storage device တွေကို `/dev` အောက်မှာ ကြည့်ရှု နိုင်ပါတယ်။

ပထမဆုံး နေရာမှာ ရှိတဲ့ hard drive ကို `/dev/sda` လို့ သတ်မှတ်ပြီး ဒုတိယနေရာမှာ ရှိတဲ့ hard drive ကို `/dev/sdb` လို့ သတ်မှတ်ပါတယ်။

တစ်ဖန်၊ ပထမဆုံး hard drive (`/dev/sda`) ရဲ့ primary partition ကို **sda1**၊ နောက်ထပ် partition ရှိသေးရင် **sda2** လို့ သတ်မှတ်ပြီး၊

ဒုတိယ hard drive (`/dev/sdb`) ရဲ့ primary partition ကို **sdb1**၊ နောက်ထပ် partition ရှိသေးရင် **sdb2** လို့ သတ်မှတ်ပါတယ်။

`ls -l` နဲ့ ခေါ်ကြည့်တဲ့အခါ permission ရဲ့ ရှေ့မှာ file ဆိုရင် - သက်တာ directory ဆိုရင် **d** သက်တာ hard drive ဆိုရင် **block device** ကို ကိုယ်စားပြုတဲ့ **b** သက်တာ ပါလေ့ရှိပါတယ်။ ဒါကို `ls -l /dev/sda` လို့ ခေါ်ကြည့်နိုင်ပါတယ်။

file system နဲ့ storage device တွေရဲ့ size၊ တပ်ဆင်ထားတဲ့ mount point တွေကို ကြည့်ရှုချင်ရင် `df` ဆိုတဲ့ command ကို **-h** ဆိုတဲ့ option နဲ့ တွဲသုံးရပါတယ်။ အောက်ပါအတိုင်းပေါ့။ `df -h` file သို့မဟုတ် directory တစ်ခုဟာ storage မာဏာ ဘယ်လောက် နေရာယူထားသလဲ သိချင်ရင် `du` ဆိုတဲ့ command ကို **-h** ဆိုတဲ့ option နဲ့ တွဲသုံးရပါတယ်။ ဥပမာ၊ `/root` ဆိုတဲ့ directory ဟာ ဘယ် size ရှိသလဲ ဘယ်လောက်နေရာ ယူထားသလဲ ကြည့်မယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ `du -h /root`

မိမိစက်မှာ storage device ဘယ်နှစ်ခု ရှိသလဲ၊ size ဘယ်လောက်လဲ၊ ဘယ်နေရာမှာ mount လုပ် ချိတ်ဆက်ထားသလဲ သိလိုရင် `lsblk` ဆိုတဲ့ command နဲ့ ခေါ်ကြည့်နိုင်ပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ `lsblk`

partition တစ်ခုရဲ့ file system သတ်မှတ်ချက်တွေကို ကြည့်ချင်ရင် `file` ဆိုတဲ့ command ကို `-sL` နဲ့ တွဲသုံးရပါတယ်။

ဥပမာ၊ ပထမဆုံး hard drive (**sda**) ရဲ့ ပထမဆုံး partition (**sda1**) ရဲ့ file system ကို ကြည့်မယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ `file -sL /dev/sda1`

partition တွေရဲ့ **UUID** (universal unique identifier) နဲ့ file system သတ်မှတ်ချက်တွေကို ကြည့်ချင်ရင် **blkid** ဆိုတဲ့ command ကို သုံးရပါတယ်။ **terminal** မှာ **blkid** လို့ ရိုက် enter ခေါက်ရုံပါပဲ။ **blkid**

storage device တွေရဲ့ file system ကို စစ်ဆေးလိုတဲ့အခါ **fsck** ဆိုတဲ့ command ကို **-N** ဆိုတဲ့ option နဲ့ တွဲပြီး အသုံးပြုရပါတယ်။

ဥပမာ၊ ပထမဆုံး hard drive (**/dev/sda**) ရဲ့ ပထမဆုံး partition (**/dev/sda1**) ကို စစ်ဆေးမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **fsck -N /dev/sda1**
#mounting_concept

cd drive တွေ **usb stick** တွေ **partition** တွေကို **/mnt/** အောက်မှာ directory တစ်ခု သတ်မှတ်ပြီး ချိတ်ဆက်ပေးထားလို့ ရပါတယ်။ ဒါကို **mount** လုပ်တယ်လို့ ခေါပါတယ်။

ဥပမာ၊ ပထမဆုံး hard drive (**sda**) ရဲ့ ပထမဆုံး partition (**sda1**) ကို **/mnt/mydata** ဆိုတဲ့ **directory** မှာ ချိတ်ဆက်မယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **mount /dev/sda1 /mnt/mydata**

ဥပမာ **cd** drive (**/dev/cdrom**) ကို **/mnt/mycd** ဆိုတဲ့ **directory** မှာ ချိတ်ဆက်မယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **mount /dev/cdrom /mnt/mycd**

mount လုပ် ချိတ်ဆက်ထားတာတွေကို ပြန်ဖြေတယ်ဆိုရင် **umount** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။

ဥပမာ **/mnt/mycd** မှာ ချိတ်ဆက်ပေးထားတဲ့ **cd drive** ကို ပြန်ဖြေတယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **umount /mnt/mycd**

ဒါလို **umount** နဲ့ ပြန်ဖြေတာ အောင်မြင်မှု ရှိမရှိ ပြတ်မပြတ် စစ်ဆေးချင်တဲ့အခါ **lsuf** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။

ဥပမာ **/mnt/mycd** မှာ ချိတ်ဆက်ပေးထားတဲ့ **cd drive** ကို **umount /mnt/mycd** ဆိုပြီး ဖြုတ်လိုက်ပြီး၊ ဒါကို ပြုတ်မပြုတ် ပြန်ကြည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **lsuf /mnt/mycd**

မိမိစက်မှာ ဘယ် **storage device** တွေ ဘယ်လို့ တပ်ဆင်ထားသလဲ သိလိုရင် **mount** ကို **grep** နဲ့ တွဲဖက် အသုံးပြုနိုင်ပါသေးတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

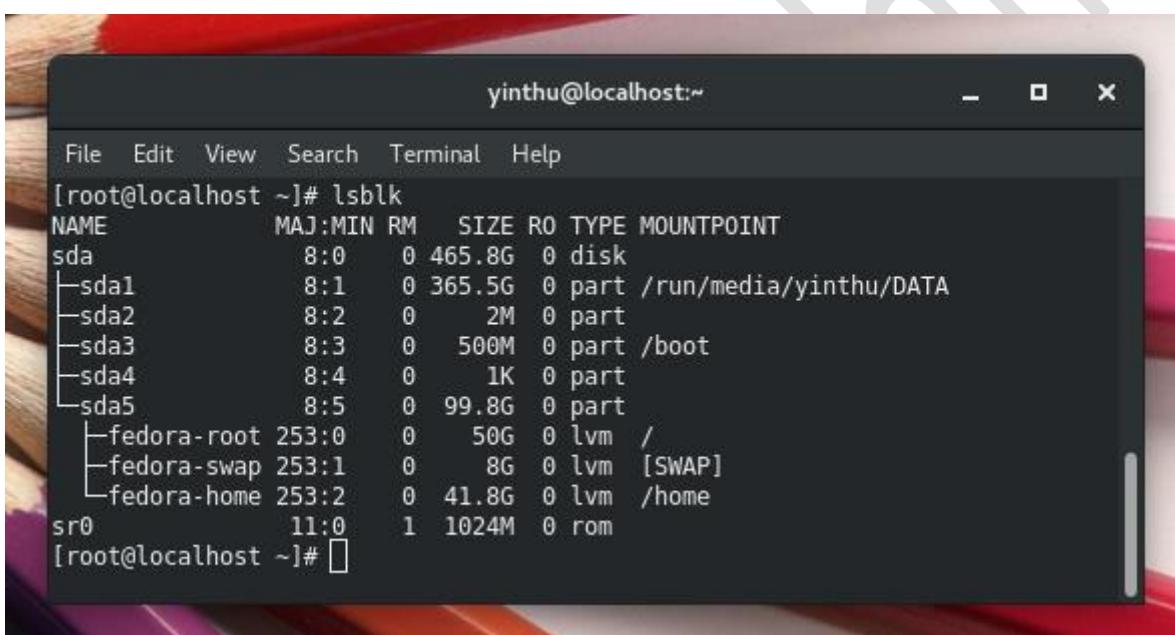
mount | grep "^/dev"

storage device တွေကို system boot လုပ်တာနဲ့ မိမိ အလိုရှိတဲ့ directory မှာ auto ချိတ်ဆက် အောင် သတ်မှတ်ပေးထားလိုပါတယ်။ ဒီလို သတ်မှတ်ဖို့အတွက်ကတော့ **/etc/fstab** ဆိုတဲ့ file ထဲမှာ ဝင်ရောက်ပြီး **source drive/partition** တွေ **mount point** တွေကို ရေးသားပေးပါတယ်။ mount နဲ့ ပတ်သက်တာတွေကို နောက်လာမယ့် သင်ခန်းစာမှာ အသေးစိတ် ရှင်းပြပါမယ်နော်။ လောလောဆယ် ဒီလောက်နဲ့ နားပါရမော့။

အားလုံး အဆင်ပြုပါမော့။

Yin Thu

(2-7-2018)



```
yinthu@localhost:~ - □ ×
File Edit View Search Terminal Help
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda        8:0    0 465.8G  0 disk
└─sda1     8:1    0 365.5G  0 part /run/media/yinthu/DATA
sda2      8:2    0   2M  0 part
sda3      8:3    0   500M  0 part /boot
sda4      8:4    0   1K  0 part
└─sda5     8:5    0  99.8G  0 part
  ├─fedora-root 253:0    0   50G  0 lvm /
  ├─fedora-swap 253:1    0   8G  0 lvm [SWAP]
  └─fedora-home 253:2    0  41.8G  0 lvm /home
sr0       11:0   1 1024M  0 rom
[root@localhost ~]#
```

how to set time with timedatectl command

linux မှာ time နဲ့ သက်ဆိုင်တာတွေကို စီမံဖို့ အတွက် date ဆိုတဲ့ command ကို အသုံးပြုတဲ့ အပိုင်းကို ယခင်က တင်ပြခဲ့ဘူးပါပြီ။ အခုတစ်ခါ linux 7 မှာ ပါဝင်လာတဲ့ **timedatectl** ဆိုတဲ့ command ကို အသုံးပြုတဲ့ အပိုင်းကို တင်ပြပါမယ်။ လေ့လာကြည့်ကြတာပေါ့။

၁။ system ရဲ့ current date နဲ့ time ကို ကြည့်ချင်ရင် **timedatectl status** ကို သုံးပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl status**

၂။ နေ့စွဲ နဲ့ အချိန်တိုက်ချင်ရင် **timedatectl** ကို **set-time** နဲ့ တွဲသုံးရပါတယ်။

ဥပမာ current date ကို 2018-07-20 ရောက်အဖြစ် သတ်မှတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl set-time "2018-07-20"**

ဥပမာ current time ကို 05:00:00 နာရီ အဖြစ် သတ်မှတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl set-time "05:00:00"**

ဥပမာ date နဲ့ time ကို တစ်ပြိုင်တည်း သတ်မှတ်ချင်တယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl set-time "2018-07-20 05:00:00"**

၃။ linux မှာ hardware clock ကို real time clock (rtc) လို့ ခေါ်ပါတယ်။ rtc ကို local timezone နဲ့ set လုပ်ချင်ရင် **set-local-rtc** နဲ့ တွဲသုံးရပါတယ်။ **set-local-rtc** မှာ **1** နဲ့ **0** ဆိုပြီး တန်ဖိုး ၂ ခု ရှိပါတယ်။ **1** ဆိုရင် **yes** ဖြစ်ပြီး **0** ဆိုရင် **no** လို့ သတ်မှတ်ပါတယ်။ rtc ကို local timezone နဲ့ set လုပ်မယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl set-local-rtc 1**

rtc ကို local timezone နဲ့ unset လုပ်မယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

timedatectl set-local-rtc 0

၄။ system ရဲ့ time ကို remote NTP Server ရဲ့ time နဲ့ synchronizing လုပ်ချင်ရင် မိမိစက်ရဲ့ network time protocol (ntp) ကို on ထားဖို့ လိုပါတယ်။ **timedatectl** ကို **set-ntp** နဲ့ တွဲသုံးရပါတယ်။ **set-ntp** မှာ **true** နဲ့ **false** ဆိုပြီး တန်ဖိုး ၂ ခု ရှိပါတယ်။ **true** ဆိုရင် **ntp** သည် **yes** ဖြစ်ပြီး **false** ဆိုရင် **ntp** သည် **no** ပါ။ ntp ကို NTP Server နဲ့ synchronize လုပ်မယ် ဆိုပါစို့။ command syntax က အောက်ပါအတိုင်းပါ။ **timedatectl set-ntp true**

ntp ကို NTP Server နဲ့ synchronize မလုပ်ဘူး ဆိုပါစို့။ command syntax က အောက်ပါအတိုင်းပါ။ **timedatectl set-ntp false**

၅။ timezone ကို ကြည့်ချင်ရင် **timedatectl** ကို **grep** နဲ့ တွဲသုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl | grep Time**

timezones တွေ အားလုံးကို ပြသပေးစေချင်ရင် **list-timezones** နဲ့ တွဲသုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **timedatectl list-timezones**

၆။ မိမိရဲ့ location နဲ့ သက်ဆိုင်တဲ့ timezone ကို ရှာဖွေလိုရင် **list-timezones** ကို **grep -o** နဲ့ တွေ့သုံးရပါတယ်။

ဥပမာ **Asia** က **Y** နဲ့စတဲ့ **timezone** ကိုရှာမယ်ဆိုပါစွဲ။ command syntax ကတေသ့ အောက်ပါ အတိုင်းပါ။ **timedatectl list-timezones | grep -o "Asia/Y.*"**

၇။ local timezone ကို **set** လုပ်ချင်ရင် **set-timezone** နဲ့ တွေ့သုံးရပါတယ်။

ဥပမာ **Asia/Yangon** ကို **set** လုပ်မယ်ဆိုပါစွဲ။ command syntax ကတေသ့ အောက်ပါအတိုင်းပါ။ **timedatectl set-timezone "Asia/Yangon"**

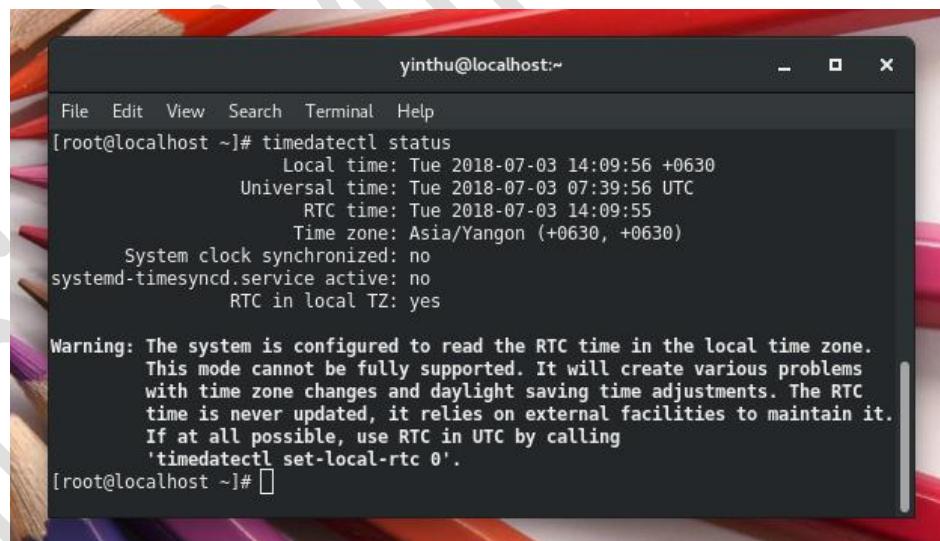
၈။ မိမိ အလိုကြရာ location အတွက် timezone ကို တစ်ဆင့်ချင်း သတ်မှတ်သွားချင်ရင် **tzselect** ကို အသုံးပြုနိုင်ပါတယ်။ command syntax ကတေသ့ အောက်ပါအတိုင်းပါ။ **tzselect**

၉။ **UTC** ကို **set** လုပ်ဖို့ command syntax ကတေသ့ အောက်ပါအတိုင်းပါ။
timedatectl set-timezone UTC

linux ပေါ်မှာ timedatectl ဆိုတဲ့ command ကို အသုံးပြုပြီး date and time နဲ့ သက်ဆိုင်တာ တွေကို စီမံတဲ့ အပိုင်းကတေသ့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစွဲ။

Yin Thu

(3-7-2018)



```
yinthu@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# timedatectl status
        Local time: Tue 2018-07-03 14:09:56 +0630
        Universal time: Tue 2018-07-03 07:39:56 UTC
                  RTC time: Tue 2018-07-03 14:09:55
                 Time zone: Asia/Yangon (+0630, +0630)
    System clock synchronized: no
systemd-timesyncd.service active: no
          RTC in local TZ: yes

Warning: The system is configured to read the RTC time in the local time zone.
This mode cannot be fully supported. It will create various problems
with time zone changes and daylight saving time adjustments. The RTC
time is never updated, it relies on external facilities to maintain it.
If at all possible, use RTC in UTC by calling
'timedatectl set-local-rtc 0'.
[root@localhost ~]#
```

linking, hard link, soft link

windows system မှာဆိုရင် file တွေ folder တွေကို shortcut ထုတ်ထားနိုင်တာ သိမယ် ထင်ပါတယ်။ linux မှာလည်း အလားတူပဲ၊ file နဲ့ directory တွေကို shortcut ပုံစံမျိုး ထုတ်ထားလို့ရပြီး၊ ဒါကို linking လုပ်တယ်လို့ သုံးနှင့်ပါတယ်။ linking မှာ -

၁။ hard link

၂။ soft link လို့မျိုးရှိပါတယ်။

၁။ **hard link** : file တွက် link ထုတ်တာကို hard link လို့ခေါ်ပါတယ်။ In ဆိုတဲ့ command ကို အသုံးပြုရဖို့

ဥပမာ current user က **tom** လို့ ထားပါတယ့်။ [/etc/named.conf](#) ဆိုတဲ့ file ကို **tom** ရဲ့ **Desktop** ဆီ **shortcut1** ဆိုတဲ့ အမည်နဲ့ link ထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **In /etc/named.conf /home/tom/Desktop/shortcut1**

ဒီနေရာမှာ တစ်ခု သိထားရမှာက link ထုတ်တော့မယ်ဆိုရင် absolute path ကို အသုံးပြုပေးရမယ် ဆိုတာပါပဲ။ ထူးခြားချက်အနေနဲ့ [/etc/](#) အောက်က [named.conf](#) ကို ဖျက်လိုက်ရင်တောင် [/home/tom/Desktop](#) အောက်က **shortcut1** မှာ **content** တွေ မပျက်စီးပဲ ကျွန်ုပါလိမ့်မယ်။ ဒါက **hard link** ရဲ့ ထူးခြားချက်ပေါ့။ **hard link** ကို **file** တွေ **link** ထုတ်ဖို့ အတွက် အသုံးပြုလို့ရ ပါတယ်။

၂။ **soft link** : directory တွက် link ထုတ်တာကို soft link လို့ ခေါ်ပါတယ်။ soft link ထုတ်ဖို့ အတွက် **In** ကို **-s** ဆိုတဲ့ option နဲ့ တွဲဖက် အသုံးပြုပေးရှုပါပဲ။

ဥပမာ current user က **tom** လို့ ထားပါတယ့်။ [/var/](#) အောက်က **log** ဆိုတဲ့ **directory** ကို **tom** ရဲ့ **Desktop** ဆီ **shortcutoflog** ဆိုတဲ့ အမည်နဲ့ link ထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။

In -s /var/log /home/tom/Desktop/shortcutoflog

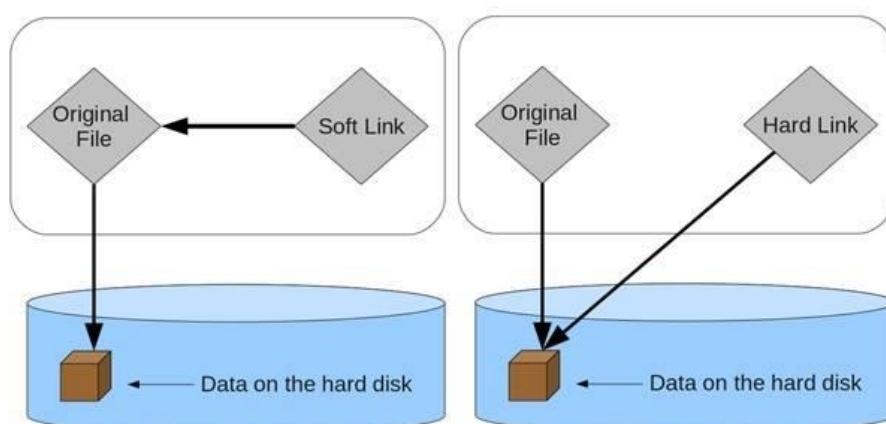
soft link မှာတော့ မူလ directory ကို ဖျက်လိုက်ရင် link ထုတ်ထားတဲ့ directory ပါ ပျက်ပါတယ်။ ဒါက soft link ရဲ့ အားနည်းချက်ပေါ့။

hard link ဖြစ်စေ soft link ဖြစ်စေ တူညီတဲ့ file system တစ်ခုတည်း ပေါ်မှာပဲ link ထုတ်လို့ ရပါတယ်။ link ထုတ်မယ့် source နဲ့ destination တို့ဟာ တူညီတဲ့ file system တစ်ခုတည်း ပေါ်မှာ ရှိနေဖို့ လိုပါတယ်။ ဆိုရရင် linux system ပေါ်မှာ file နဲ့ directory တွေကို linking လုပ်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(4-7-2018)



finding via linux terminal

တင်ဖူးသမျှ ရှည်လျားပြီး လက်ဝင်တဲ့ post တစ်ခုပါပဲ။ ကဲ ! စရအောင် windows operating system မှာ file တွေ folder တွေကို search နဲ့ ရှာဖွေနိုင်တာ သိလိမ့်မယ်လို့ ယူဆပါတယ်။ linux မှာလည်း အလားတူ GUI ကနေ သက်သက်သာသာ ရှာဖွေနိုင်ပါတယ်။ အဲ ! terminal ကနေ တစ်ဆင့် ဆိုရင်တော့ find ဆိုတဲ့ command ကို အသုံးပြုပြီး ရှာဖွေနိုင်ပါတယ်။ ဒီလို ရှာဖွေရာမှာ ->

၁။ finding with name

၂။ finding based on type

၃။ finding based on their permissions

၄။ finding based on owner and group

၅။ finding based on date and time

၆။ finding based on size ဆိုပြီး ၆ ပိုင်း ရှိပါတယ်။

ရှုပ်ထွေးသယောင် ရှိပေမယ့် တကယ်တော့ ရိုးရှင်းပါတယ်။ လေ့လာကြည့်ရအောင်။

၁။ finding with name

file တစ်ခုကို ရှာဖို့အတွက် **-name** ဆိုတဲ့ option ကို တွဲသုံးရပါတယ်။

ဥပမာ **sshd_config** ဆိုတဲ့ **file** ကို **root directory /** ကနေ စရာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax အောက်ပါအတိုင်းပါ။ **find / -name sshd_config**

file တွေကို ရှာဖွေရာမှာ အမည် အတိအကျမသိရင် **wild card *** တွေနဲ့တွဲဖက်ပြီး ရှာဖွေနိုင်ပါတယ်။ ဥပမာ .txt နဲ့ ဆုံးတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစို့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -name "*.txt"**

ဥပမာ **/etc/** အောက်မှာ **passwd** ဆိုတဲ့ **file** ကို ရှာမယ်။ ဒါပေမယ့် **file** ကို အကုန် မမှတ်မိဘူး။

pass ပဲ မှတ်မိတယ် ဆိုပါစို့။ အောက်ပါအတိုင်း **wild card *** နဲ့ တွဲဖက်ပြီး ရှာဖွေနိုင်ပါတယ်။

find /etc -name "*pass*" file တွေကို ရှာဖွေရာမှာ capital letter, small letter မခဲ့ခြားပဲ ရှာဖွေမယ်ဆို **-i** ဆိုတဲ့ option ကို အသုံးပြုရပါတယ်။

ဥပမာ **/var/** အောက်မှာ **messages** ဆိုတဲ့ အမည်ပါတဲ့ **file** တွေကို ရှာချင်တယ်။ စာလုံး အကြီး အသေး မခဲ့ခြားဘူး။ အမည်ကိုလည်း တိတိကျကျ မသိဘူးဆိုပါစို့။ **-i** နဲ့ **wild card *** ကို တွဲဖက်ပြီး အောက်ပါအတိုင်း ရှာဖွေနိုင်ပါတယ်။ **find /var -i -name *messages***

file အမည်တွေကို အခြေပြုပြီး **find** နဲ့ ရှာဖွေတဲ့ နည်းလမ်းတွေကတော့ ဒီလောက်ပါပဲ။ အသုံးလည်း များပါတယ်နော်။

JII finding based on type

type ကို အခြေပြုပြီး ရှာဖွေတဲ့အခါ -**type** option ကို အမျိုးအစားကိုကိုယ်စားပြုတဲ့ ထပ်တိုး option ၅ မျိုးနဲ့ တွဲဖက် အသုံးပြုရပါတယ်။ -**type** နဲ့ တွဲဖက် အသုံးပြုရမယ့် ထပ်တိုး option တွေကတော့ ->

၁။ **file** ကို ကိုယ်စားပြုတဲ့ **f**

JII **directory** ကို ကိုယ်စားပြုတဲ့ **d**

၃။ **soft link** ကို ကိုယ်စားပြုတဲ့ **I**

၄။ **hard link** ကို ကိုယ်စားပြုတဲ့ **f -links**

၅။ **block device** ကို ကိုယ်စားပြုတဲ့ **b** တို့ပဲ ဖြစ်ပါတယ်။

ဥပမာ / ကနေ စပြီး **.php** နဲ့ဆုံးတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -name -type f "*.php"**

ဥပမာ /etc အောက်မှာ **sysconfig** ဆိုတဲ့ **directory** ကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find /etc -name -type d sysconfig**

ဥပမာ / အောက်ကနေ စပြီး **empty** ဖြစ်နေတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -type f -empty**

ဥပမာ / အောက်ကနေ စပြီး **empty** ဖြစ်နေတဲ့ **directory** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -type d -empty**

ဥပမာ / အောက်ကနေစပြီး **soft link** ချိတ်ထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type l (အယ်လ်အသေး)**

ဥပမာ / အောက်ကနေစပြီး **hard link** ၁ ခုထက် ပိုချိတ်ထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type f -links +1**

ဥပမာ /dev အောက်က **storage block device** တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find /dev -type b**

type ကို အခြေပြုပြီး find နဲ့ ရှာဖွေတဲ့ နည်းလမ်းတွေကတော့ ဒီလောက်ပါပဲ။

၃။ finding based on their permission

ဒါကတော့ ရှင်းပါတယ် မိမိ ရှာလိုတဲ့ **file** ကို **permission** ပေါ်မှတည်ပြီး ရှာဖွေတာပါ။ -**perm** ဆိုတဲ့ option ကို အသုံးပြုရပါတယ်။

ဥပမာ / အောက်ကနေ စပြီး **permission 750** ပေးထားတဲ့ file တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type f -perm 750**

ဥပမာ / အောက်ကနေ စပြီး **permission 750** ပေးထားတဲ့ **directory** တွေကို ရှာမယ် ဆိုပါစွဲ။
အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type d -perm 750**

permission ပေါ်မှုတည်ပြီး ရှာဖွေရမှာ ဖော်ပြထားတဲ့ **permission** နဲ့ ဆန့်ကျင်ဘက် **file** ကို
ရှာချင်ရင် ! ဆိုတဲ့ ထပ်တိုး option လေးကို ထည့်ပေးရပါတယ်။

ဥပမာ / အောက်ကနေ စပြီး **permission 750** မဟုတ်တဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံး
ပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type f -perm ! 750**

ဥပမာ / အောက်ကနေ စပြီး **permission 750** မဟုတ်တဲ့ **directory** တွေကို ရှာမယ် ဆိုပါစွဲ။
အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -type d -perm ! 750**

ဥပမာ **user** အတွက် **special permission** ပေးထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြု
ရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -perm /u=s**

ဥပမာ **group** အတွက် **special permission** ပေးထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြု
ရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -perm /g=s**

ဥပမာ **other** အတွက် **special permission** ပေးထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြု
ရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -perm /o=t**

ဥပမာ **user** အတွက် **read only permission** ပေးထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြု
ရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -perm /u=r**

ဥပမာ **group** အတွက် **read only permission** ပေးထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။
အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -perm /g=r**

permission ကို အခြေပြုပြီး **find** နဲ့ ရှာဖွေတဲ့ နည်းလမ်းတွေကောင်တော့ ဒီလောက်ပါပဲ။

၄။ finding based on owner and group

ဒါကတော့ **user** တွေ **group** တွေ **user id (uid)** တွေ **group id (gid)** တွေပေါ်မှုတည်ပြီး ရှာဖွေ
တဲ့ သဘောပါ။

ဥပမာ **student** ဆိုတဲ့ **user** ပိုင်တဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax
ကတော့ အောက်ပါအတိုင်းပါ။ **find -user student**

ဥပမာ **user id 1000** က ပိုင်ဆိုင်တဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command
syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -uid 1000**

ဥပမာ **hacking** ဆိုတဲ့ **group** ထဲမှာ ပါတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command
syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -group hacking**

ဥပမာ **group id 1000** က ပိုင်ဆိုင်တဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command
syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -gid 1000**

ဥပမာ **student** ဆိုတဲ့ **user** နဲ့ **hacking** ဆိုတဲ့ **group** က ပိုင်တဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။
အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

find -user student -group hacking

user နဲ့ group ပေါ်မှတည်ပြီး ရှာဖွေတဲ့ နည်းလမ်းတွေကေတော့ ဒီလောက်ပါပဲ။

၅။ finding based on date and time

ဒါကတော့ ရက်စွဲနဲ့ အချိန်ပေါ်မှတည်ပြီး ရှာဖွေတဲ့ နည်းလမ်းပေါ့။

ဥပမာ ပြီးခဲ့တဲ့ ရက် ၅၀ က **modify** လုပ်ထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ **-mtime** ဆိုတဲ့ option ကို သုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -mtime 50**

ဥပမာ ပြီးခဲ့တဲ့ ရက် ၅၀ က **access** လုပ်ခံထားရတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ **-atime** ဆိုတဲ့ option ကို သုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -atime 50**

ဥပမာ ပြီးခဲ့တဲ့ ရက် ၅၀ နဲ့ ၁၀၀ ကြားက **modify** လုပ်ထားတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ **-mtime** ဆိုတဲ့ option ကိုပဲ သုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

find -mtime +50 -mtime -100

ဥပမာ ပြီးခဲ့တဲ့ ၁ နာရီက **modify** လုပ်ခံထားရတဲ့ file တွေကို ရှာမယ် ဆိုပါစွဲ။ **-mmin** ဆိုတဲ့ option ကို သုံးရပါတယ်။ **m** က **modify .. min** က **minute** ပေါ့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -mmin -60**

ဥပမာ ပြီးခဲ့တဲ့ ၁ နာရီက **access** လုပ်ခံထားရတဲ့ file တွေကို ရှာမယ် ဆိုပါစွဲ။ **-amin** ဆိုတဲ့ option ကို သုံးရပါတယ်။ **a** က **access .. min** က **minute** ပေါ့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -amin -60**

ဥပမာ ပြီးခဲ့တဲ့ ၁ နာရီက **change** လုပ်ခံထားရတဲ့ file တွေကို ရှာမယ် ဆိုပါစွဲ။ **-cmin** ဆိုတဲ့ option ကို သုံးရပါတယ်။ **c** က **change .. min** က **minute** ပေါ့။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find -cmin -60**

date and time ကို အခြေပြုပြီး ရှာဖွေတဲ့ နည်းလမ်းတွေကေတော့ ဒီလောက်ပါပဲ။

၆။ finding based on size

size ပေါ်မှတည်ပြီး ရှာဖွေတဲ့အခါ အားလုံး သတ်မှတ်ချက် ၃ မျိုးကို ထည့်သွင်းပေးရပါတယ်။ size သတ်မှတ်ချက် ၃ မျိုးကတော့ -

၁။ **K** (kilobyte) ၂။ **M** (megabyte) ၃။ **G** (gigabyte) တို့ပါပဲ။

ဥပမာ ၅၀ **megabyte** ရှိတဲ့ **file** တွေကို ရှာမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -size 50M**

ဥပမာ **50 megabyte** နဲ့ **100 megabyte** ကြား ရှိတဲ့ file တွေကို ရှာမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **find / -size +50M -size -100M**

size ပေါ်မှုတည်ပြီး find နဲ့ ရှာဖွေတဲ့ နည်းလမ်းတွေကေတော့ ဒီလောက်ပါပဲ။ find အသုံးပြုပြီး ရှာဖွေတဲ့ နည်းလမ်းတွေကို တင်ဆက်ပြီးပါပြီ။ ဒီ ရှာဖွေနည်းတွေအပြင် file တွေကို ရှာဖွေပြီးရလာတဲ့ file တွေကို ဖျက်ပစ်နိုင်ပါသေးတယ်။ **-exec rm {} \;** လေးကို ထည့်သုံးပေးရှုပါပဲ။

ဥပမာ **/home** ဆိုတဲ့ directory အောက်က **.mp3** နဲ့ ဆုံးတဲ့ **10 megabyte** ကျော်တဲ့ **file** တွေကို ရှာပြီး တန်းဖျက်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

find /home -type -f -name -size +10M "*.mp3" -exec rm {} \;

find နဲ့ သက်ဆိုင်တဲ့အကြောင်းတွေကတော့ ဒီလောက်ပါပဲ။ ဖြည့်ဖြည့်ချင်း မှတ်သားပြီး ဖတ်သွားရင် နားလည်ရ လွယ်ကူသွားမှုပါ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(5-7-2018)



controlling jobs

အသုံးနဲ့ပေါ်မယ့် သိထားသင့်တဲ့ topic တစ်ခုပါ။ အရေးအကြောင်း လိုအပ်လာတဲ့အခါ အသုံးချလို့ ရတာပေါ့နော်။ linux ရဲ့ terminal ထဲမှာ command တစ်ခုကို run လိုက်တဲ့အခါ job ၁ ခု ဖြစ်ပေါ်လာပါတယ်။ အဲဒီ job တစ်ခုမှာ -->

၁။ **job id**

၂။ **process id**

၃။ **state** နဲ့

၄။ **job** ဆိုပြီး အပိုင်း ၄ ခု ရှိပါတယ်။

ဥပမာ **sleep 1500** လို့ ရှိက်ပြီး shell ကို **sleep mode** ပို့လိုက်တယ် ဆိုပါစို့။ ဒါဆို **sleep 1500**

သည် job တစ်ခု ဖြစ်သွားပါပြီ။ ဒီ **job .. run** နေစဉ်မှာ shell ပေါ်မှာ တွေား **command run** လို့ မရပါဘူး။

၁။ ဒီ **job** ကို **kill** ချင်ရင် keyboard ပေါ်က **ctrl+c** ကို နိုင်ပြီး **kill** နိုင်ပါတယ်။

၂။ နောက်တစ်ခုက၊ ဒီ **job** ကို မရှင်းလင်းပဲ **sleep 1500** ဆိုတဲ့ **job** ကို keyboard ပေါ်က **ctrl+z** ကို နိုင်၊ **stop** လုပ်ပြီး **background** ကို ပို့ထားလို့ ရပါတယ်။ ဒါဆို ဒီ **job** ရဲ့ **state** ဟာ **stop** ဖြစ်ပြီး background ကို ရောက်သွားပါလိမ့်မယ်။ **ctrl+z**

၃။ **ctrl+z** နှင့်လိုက်လို့ **background** ကို ရောက်ပြီး **stop** ဖြစ်နေတဲ့ **job** တစ်ခုကို **background** မှာ **run** နေစေချင်ရင် **bg %jobID** ဆိုတဲ့ format ကို သုံးပြီး run နိုင်ပါတယ်။

ဥပမာ **background** မှာ ရှိပြီး **stop** ဖြစ်နေတဲ့ **sleep 1500** ဆိုတဲ့ **job** ရဲ့ **id** က ၁ ဆိုပါစို့။
အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **bg %1**

၄။ **ctrl+z** နှင့်လိုက်လို့ **background** ကို ရောက်ပြီး **stop** ဖြစ်နေတဲ့ **job** တစ်ခုကို **foreground** ဆီပြန်ခေါ်ပြီး **run** နေစေချင်ရင် **fg %jobID** ဆိုတဲ့ format ကို သုံးပြီး run နိုင်ပါတယ်။

ဥပမာ **sleep 1500** ဆိုတဲ့ **job** ရဲ့ **id** က ၁ ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **fg %1**

၅။ **command** တစ်ခုကို စစ် **run** ချင်းမှာပဲ **background** ကို ပိုပြီး **run** နေစေချင်ရင် & ဆိုတဲ့ သက်တကို **command** နောက်မှာ ထည့်ပေးရှုနဲ့ အဆင်ပြေပါတယ်။

ဥပမာ **sleep 1500** ဆိုတဲ့ **command** ကို စစ် **run** ချင်း **background** မှာသွား **run** စေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။ **sleep 1500 &**

၆။ ဒီလို **ctrl+z** နှင့်လိုက်လို့ **background** ကို ရောက်သွားတဲ့ **job** တွေရဲ့ အခြေအနေကို **jobs** ဆိုတဲ့ command နဲ့ ခေါ်ကြည့်နိုင်ပါတယ်။

jobs -l လို့ ရှိက် enter ခေါက်လိုက်ရင် **job id, process id, run** နေလား **stop** ဖြစ်နေလားဆိုတဲ့ **state** (အခြေအနေ), **job** အမျိုးအစား စတာတွေကို တွေ့မြင်ရပါလိမ့်မယ်။

ဥပမာ **background** မှာရှိနေတဲ့ **job** အားလုံးရဲ့ သတ်မှတ်ချက်တွေကို ကြည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **jobs -l**

ဥပမာ **background** မှာ **run** နေတဲ့ **job** တွေရဲ့အခြေအနေ သတ်မှတ်ချက်တွေကို သီးသန့် ကြည့်မယ်ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **jobs -r**

ဥပမာ **background** မှာ **stop** ဖြစ်နေတဲ့ **job** တွေရဲ့အခြေအနေ သတ်မှတ်ချက်တွေကို သီးသန့် ကြည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **jobs -s**

စာလည်း နည်းနည်း ရှည်သွားပါပြီ။ terminal ကနေ **jobs** တွေကို ကိုင်တွယ်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(6-7-2018)

```
nixcraft@wks05:~$ jobs -l
[1] 7895 Running
[2] 7906 Running
[3]- 7910 Running
[4]+ 7946 Stopped
nixcraft@wks05:~$
```

jobID	PID	State	Jobs
[1]	7895	Running	gpass &
[2]	7906	Running	gnome-calculator &
[3]-	7910	Running	gedit fetch-stock-prices.py &
[4]+	7946	Stopped	ping cyberciti.biz

controlling processes

ဒီတစ်ခါကတော့ linux system ပေါ်မှာ run နေတဲ့ process တွေကို ကိုင်တွယ်မယ့် အကြောင်းပါ။ ဒီအပိုင်းက လွယ်ပေါ်မယ့် အရေးကြီးသလို ကျဉ်းထဲကျပ်တည်းကျရင် တကယ်လည်း သုံးရပါတယ် နော်။ linux ပေါ်မှာ run နေတဲ့ process တွေကို ငါးတို့ရဲ့ process id ကနေတစ်ဆင့် ထိန်းချုပ်ရ ပါတယ်။ run နေတဲ့ process တွေနဲ့ process id တွေကို **process -aux** ဆိုတဲ့ command နဲ့ ခေါ်ကြည့်နိုင်ပါတယ်။ **ps -aux** လို့ ရိုက် enter ခေါက်လိုက်ရင် process တွေနဲ့ သူတို့ရဲ့ id တွေ ကျလာပါလိမ့်မယ်။ process တစ်ခုကို ထိန်းချုပ်ဖို့ အတွက် -

၁။ kill

၂။ killall

၃။ pkill

၄။ pgrep

၅။ pstree စတဲ့ command တွေကို အသုံးပြုရပါတယ်။ ဒီအထဲကမ အသုံး အများဆုံးက kill ပါ။ process တွေကို ထိန်းချုပ်ရမှာ အရေးပါတဲ့ options တွေနဲ့ သူတို့ရဲ့ code no. တွေကတော့ -

(9) SIGKILL

(15) SIGTERM

(18) SIGCONT

(19) SIGSTOP တို့ပါပဲ။ ဒီအထဲက အသုံး အများဆုံးက **9** ပါ။

ဥပမာ **process id 1500** ဆိုတဲ့ **process** တစ်ခုကို **KILL** မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **kill -SIGKILL 1500** သို့မဟုတ် **kill -9 1500**

ဥပမာ **process id 1500** ဆိုတဲ့ **process** တစ်ခုကို **TERMINATE** လုပ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

kill -SIGTERM 1500 သို့မဟုတ် **kill -15 1500**

ဥပမာ **process id 1500** ဆိုတဲ့ **process** တစ်ခုကို **CONTINUE** နဲ့ ဆက် **run** မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

kill -SIGCONT 1500 သို့မဟုတ် **kill -18 1500**

ဥပမာ **process id 1500** ဆိုတဲ့ **process** တစ်ခုကို **STOP** လုပ် ရပ်ထားမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

kill -SIGSTOP 1500 သို့မဟုတ် **kill -19 1500**

ဒုံးအပြင် လက်ရှိ **account** မှာ **run** နေတဲ့ **process** တွေသာမက မိမိစက်ကို **ssh** ကနေ တစ်ဆင့် ဖြစ်စေဝင်ရောက် အသုံးပြုနေတဲ့ **user** တွေရဲ့ **process** တွေကိုပါ ထိန်းချုပ်နိုင်ပါသေးတယ်။ မိမိ

စက်ကို ဘယ်သူတွေ **login** ဝင်နေသလဲဆိုတာကိုတော့ **w -f** ဆိုတဲ့ command နဲ့ ကြည့်ရှုနိုင်ပါတယ်။

ဥပမာ **w -f** နဲ့ ကြည့်လိုက်တဲ့အခါ **tom** ဆိုတဲ့ **user** ဟာ မိမိ စက်ကို **login** ဝင်ထားတယ်။ **tom** ရဲ့ **process** တွေကို ကြည့်မယ်ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **pgrep -l -u tom**

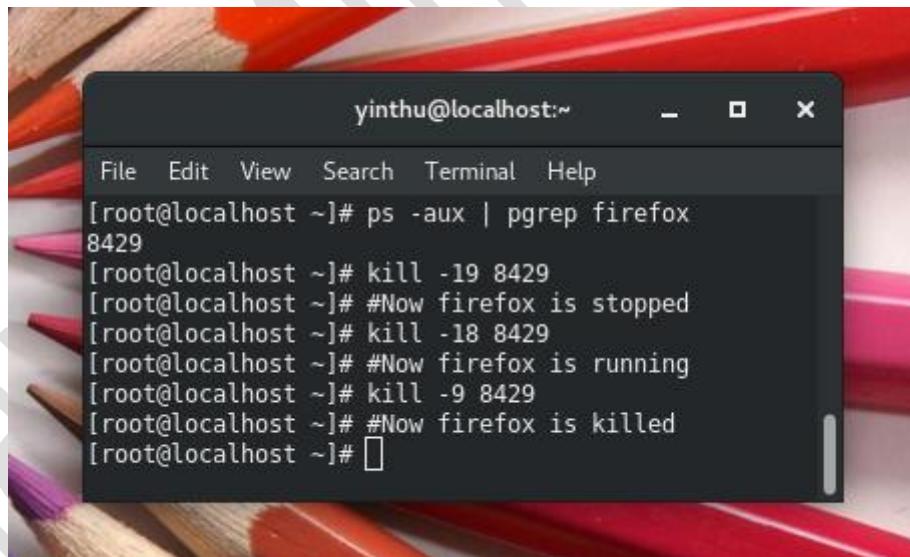
မိမိ စက်ကို **login** ဝင်နေတဲ့ **user** ရဲ့ **process** တွေကို **pgrep** အပြင် **pstree** ဆိုတဲ့ command သုံးပြီးလည်း ကြည့်နိုင်ပါသေးတယ်။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **pstree -p tom**

tom ဆိုတဲ့ **user** ဆီမှာ **run** နေတဲ့ **process id 528** ကို လှမ်း **kill** မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **pkill -u tom -SIGKILL 528**

kill နဲ့ သက်ဆိုင်တဲ့ တစ်ခြား options တွေကို ဆက်လက်လေ့လာလိုရင် **kill -l** လို့ ရှိက် enter ခေါက်ပြီး လေ့လာနိုင်ပါတယ်။ linux system ပေါ်မှာ process တွေကို ထိန်းချုပ်တဲ့ အပိုင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(9-7-2018)



A screenshot of a terminal window titled "yinthu@localhost:~". The window shows a series of commands being run in a terminal session. The user runs "ps -aux | pgrep firefox" to find the process ID (8429). Then, they run "kill -19 8429" to stop the process. After stopping it, they run "kill -18 8429" to start it again. Finally, they run "kill -9 8429" to kill the process completely. The terminal window has a dark background and light-colored text.

```
[root@localhost ~]# ps -aux | pgrep firefox
8429
[root@localhost ~]# kill -19 8429
[root@localhost ~]# #Now firefox is stopped
[root@localhost ~]# kill -18 8429
[root@localhost ~]# #Now firefox is running
[root@localhost ~]# kill -9 8429
[root@localhost ~]# #Now firefox is killed
[root@localhost ~]#
```

monitoring the system with top

windows မှာဆိုရင် system ကို monitoring လုပ်ဖို့ task manager ပါတာ သိကြမှာပါ။ အလားတူပါပဲ linux မှာ top ဆိုတဲ့ command ကို အသုံးပြုပြီး system, cpu usage, memory အခြေအနေ နဲ့ process တွေကို live စောင့်ကြည့်နိုင်ပါတယ်။ စလိုက်ရအောင် !

၁။ system နဲ့ process အားလုံးကို ကြည့်ချင်ရင် terminal မှာ top လို့ ရှိက် enter ခေါက်ရုပါပဲ။ command syntax ကတော့ အောက်ပါ အတိုင်းပါ။ **top**

J။ process id တွကိုပဲ သီးသန့်ကြည့်မယ်ဆိုရင် -p ဆိုတဲ့ option နဲ့ တွဲသုံးရပါတယ်။ top -p ၃။ user တစ်ဦးရဲ့ process တွကိုပဲ ကြည့်ချင်တဲ့ အခါ -u ဆိုတဲ့ option နဲ့ တွဲသုံးရပါတယ်။ ဥပမာ **tom** ဆိုတဲ့ **user** ရဲ့ **process** တွကိုပဲ သီးသန့်ကြည့်မယ် ဆိုပါစဲ။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။ top -u tom

ငဲ့ **idle** ဖြစ်နေတဲ့ **process** တွကို မကြည့်ဘူး **active** ဖြစ်နေတဲ့ **process** တွကိုပဲ ကြည့်မယ် ဆိုရင် -i ဆိုတဲ့ option ကို တွဲသုံးရပါတယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

top -i

၅။ **top** နဲ့ ဖော်ပြထားတဲ့ **result** တွကို သိမ်းထားချင်ရင် **keyboard** ပေါက **shift+w** ကို တွဲနှပ်ပေးရပါတယ်။ ဒီအခါ **result** တွကို </etc/> အောက်မှာ **.toprc** ဆိုတဲ့ **file** အဖြစ် **save** လုပ်သွားပါလိမ့်မယ်။ </etc/.toprc>

top နဲ့ ကြည့်ရှုနေရာကနေ ထွက်ချင်ရင်တော့ **keyboard** ပေါက **q** ကို နှိပ်ပြီး ပြန်ထွက်နိုင်ပါတယ်။

top ကို အသုံးပြုပြီး system နဲ့ process တွကို monitoring လုပ်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(11-7-2018)

```

yinthu@localhost:~ - □ ×
File Edit View Search Terminal Help
top - 10:06:17 up 1:09, 1 user, load average: 0.74, 1.05, 1.17
Tasks: 238 total, 1 running, 176 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.0 us, 1.2 sy, 0.0 ni, 96.4 id, 0.1 wa, 0.3 hi, 0.1 si, 0.0 st
KiB Mem : 3936108 total, 961692 free, 1730196 used, 1244220 buff/cache
KiB Swap: 8388604 total, 8388604 free, 0 used. 1892400 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1477 yinthu 20 0 4047680 236868 97700 S 3.7 6.0 35:10.10 gnome-shell
5283 yinthu 20 0 1376960 121444 68636 S 3.3 3.1 0:09.81 chrome
1387 yinthu 20 0 453268 89448 55076 S 1.7 2.3 0:20.09 Xorg
2268 yinthu 20 0 829668 37680 27764 S 1.0 1.0 0:02.40 gnome-term+
733 root -51 0 0 0 0 S 0.7 0.0 0:15.16 irq/51-nvi+
2580 yinthu 20 0 1386968 195112 117460 S 0.7 5.0 0:54.15 chrome
5063 yinthu 20 0 4166596 417540 94896 S 0.7 10.6 2:43.10 chrome
1 root 20 0 236796 9708 6852 S 0.3 0.2 0:02.41 systemd
731 root 20 0 0 0 0 S 0.3 0.0 0:00.63 nvidia-mod+
4066 root 20 0 0 0 0 I 0.3 0.0 0:00.33 kworker/1:2
5380 root 20 0 257348 4324 3436 R 0.3 0.1 0:00.12 top
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:+
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_+
8 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
9 root 20 0 0 0 0 I 0.0 0.0 0:00.90 rcu_sched

```

grep ဖြင့် စာသားများကို ဆွဲထုတ်ခြင်း

log file တွေ **text file** တွေ **process output** တွေထက် စာသားတွေကို ဆွဲထုတ်ချင်တဲ့အခါ **grep** command ကို အသုံးပြုရပါတယ်။ အရင်ဆုံး linux machine ကို root account နဲ့ login ဝင်ပါ။ current user's home directory အောက်မှာ **touch file1** နဲ့ **file1** လေး ၁ ခု ဆောက်လိုက်ပါ။

touch file1

vi file1 command နဲ့ **file1** ကို vim text editor နဲ့ ဖွင့်ပြီး အောက်က စာကြောင်းလေးတွေကို ရှိက်ထည့်ပါ။

sky

fly

cat

table

blue

ကဲ ! ခုခွဲ့ **grep** ကို စမ်းသပ်ဖို့ အသင့်ဖြစ်နေပါပြီ။

၁။ စာလုံးအတိအကျပ်မူတည်ပြီး စာကြောင်းကို ဆွဲထုတ်ချင်တဲ့အခါ grep ရှုံးမှာ စာလုံး အတိ အကျကို expression အနေနဲ့ ထည့်ပေးရပါတယ်။

ဥပမာ **file1** ထဲက **cat** ဆိုတဲ့ စာလုံးကို ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep "cat" file1**

၂။ ထိပ်ဆုံး စာလုံးပေါ်မူတည်ပြီး ဆွဲထုတ်လိုတဲ့အခါ ^ သက်တကို အသုံးပြုရပါတယ်။

ဥပမာ **c** နဲ့ စတဲ့ စာကြောင်းကို ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါမတိုင်းပါ။ **grep "^c" file1**

၃။ နောက်ဆုံး စာလုံးပေါ်မူတည်ပြီး ဆွဲထုတ်လိုတဲ့အခါ \$ သက်တကို အသုံးပြုရပါတယ်။

ဥပမာ **t** နဲ့ ဆုံးတဲ့ စာကြောင်းကို ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါမတိုင်းပါ။ **grep "t\$" file1**

၄။ expression ထဲမှာ . * စတဲ့ wild card တွေကို အသုံးပြုပြီး စာကြောင်းတွေကို ဆွဲထုတ် နိုင်ပါတယ်။

ဥပမာ **table** ကို **wild card** သုံးပြီး ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep "t.*e" file1**

၅။ အစ စာလုံးနဲ့ အဆုံး စာလုံး ကြားက စာလုံးတွေကို ခန့်မှန်းပြီး ဆွဲထုတ်လိုတဲ့အခါ [] ကို အသုံးပြုရပါတယ်။

ဥပမာ **table** ဆိုတဲ့ စာလုံးမှာ **t** နဲ့ **e** ကြားက စာလုံးတွေကို ခန့်မန်းပြီး ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep "t[!ba]e" file1**

၆။ အစ စာလုံးနဲ့ အဆုံး စာလုံး ကြားက စာလုံး အရေအတွက်ပေါ်မှုတည်ပြီး ဆွဲထုတ်ချင်တဲ့အခါ .၏၏ ကို အသုံးပြုရပါတယ်။

ဥပမာ **table** ဆိုတဲ့ စာလုံးမှာ **t** နဲ့ **e** ကြားက အရေအတွက်ပေါ်မှုတည်ပြီး ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep "t.\{3\}e" file1**

၇။ ဆွဲထုတ်လိုက်တဲ့ စာသားနဲ့ သူ့အောက်က ၂ ကြောင်းကိုပါ ကြည့်ချင်ရင် **-A2** ဆိုတဲ့ option ကို သုံးရပါတယ်။

ဥပမာ **cat** နဲ့ သူ့အောက်က ၂ ကြောင်းကိုပါ ကြည့်ချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -A2 "cat" file1**

၈။ ဆွဲထုတ်လိုက်တဲ့ စာသားနဲ့ သူ့အပေါ်က ၂ ကြောင်းကိုပါ ကြည့်ချင်ရင် **-B2** ဆိုတဲ့ option ကို သုံးရပါတယ်။

ဥပမာ **cat** နဲ့ သူ့အပေါ်က ၂ ကြောင်းကိုပါ ကြည့်ချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -B2 "cat" file1**

၉။ စာလုံးကို **case-sensitive** မဖြစ်ပဲ ဆွဲထုတ်လိုတဲ့အခါ **-i** ဆိုတဲ့ option ကို အသုံးပြုရပါတယ်။

ဥပမာ **cat** ကို **case-sensitive** မဖြစ်စေပဲ ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -i "CAT" file1**

၁၀။ ဆွဲထုတ်လိုတဲ့ စာသားကို current directory အောက်မှာ ရှိတဲ့ file တိုင်းကနေ ရှာဖွေ ဆွဲထုတ်လိုတဲ့အခါ **-r** ဆိုတဲ့ option ကို အသုံးပြုရပါတယ်။

ဥပမာ **cat** ကို file တိုင်းကနေ ရှာဖွေ ဆွဲထုတ်ပေးစေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -r "cat"**

၁၁။ ဖော်ပြလိုက်တဲ့ expression မပါတဲ့ စာကြောင်းတွေကိုချည်း ဆွဲထုတ်လိုတဲ့အခါ **-v** ဆိုတဲ့ option ကို အသုံးပြုရပါတယ်။

ဥပမာ **cat** မပါတဲ့ စာကြောင်းတွေကိုချည်း ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -v "cat" file1**

၁၂။ expression ကို ၁ ခုထက် ပိုဖော်ပြပြီး စာကြောင်းတွေကို ဆွဲထုတ်လိုတဲ့အခါ **-e** ဆိုတဲ့ option ခံပေးရပါတယ်။

ဥပမာ cat နဲ့ table ဆိုတဲ့ စာကြောင်း ၂ ခုကို တစ်ပြိုင်နက် ဆွဲထုတ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **grep -e "cat" -e "table" file1**

ကဲ ! လွယ်ပါတယ်နော်။

grep ကို လက်တွေ့သုံးပြရရင် **cpu info** ကို ကြည့်မယ် ဆိုပါစိုး။

cat /proc/cpuinfo | grep -i "model"

အဲလို သုံးရပါတယ်နော်။ **grep** ကို အသုံးပြုပြီး စာသားတွေကို ဆွဲထုတ်တဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံးပဲ အဆင်ပြေပါစေ။

Yin Thu

(13-7-2018)



```
root@dns:~  
login as: root  
root@192.168.1.163's password:  
Last login: Wed Feb 20 14:49:46 2019 from 192.168.1.103  
[root@dns ~]# cat /proc/cpuinfo | grep -i "model"  
model : 60  
model name : Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz  
[root@dns ~]#
```

at ကို အသုံးပြု၍ job တစ်ခုကို အချိန်ယေားဖြင့် run စေခြင်း

linux ပေါ်မှာ job တစ်ခုကို အချိန်ယေားလေး ဆွဲပြီး run ခိုင်းမယ်။ နောက် ၁ နာရီ နေရင်၊ နောက် ၅ မိနစ် နေရင်၊ နောက် ၁ ရက် နေရင်၊ စသဖြင့်ပေါ့။ ဒီလို run စေချင်တဲ့အခါ **at** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ package name က **at** ဖြစ်ပြီး သူ့ရဲ့ service name က **atd** ပါ။

systemctl start atd **systemctl enable atd**

at ကို အသုံးပြုပြီး နေစဉ် လစဉ် ပုံမှန်တော့ run ခိုင်းလို့ မရပါဘူး။ one time ပဲ ရပါတယ်။ ဒါကြောင့် သူကို **one time task scheduling** လို့ ခေါ်ပါတယ်။ **at** ကို အသုံးတော့ နဲပါတယ်နော်။ ၁။ ဥပမာ 1 job တစ်ခု ကို နောက် ၅ မိနစ် နေရင် run စေချင်တယ် ဆိုပါစိုး။ အသုံးပြုရမယ့် command syntax က အောက်ပါအတိုင်းပါ။

at now+5min လို့ ရိုက် enter ခေါက်ပါ။ ဒါအခါ **at >** ဆို ပေါ်လာပြီး **at > shell** ထဲကို ရောက်ရှိသွားပါလိမ့်မယ်။ အဲဒါ shell ထဲမှာ ကိုယ် run စေချင်တဲ့ **job** ကို ရိုက်ထည့် ပေးရမှာပါ။

at > echo "hello world" >> ~/file1

ခုလို ရိုက်ထည့်ပြီးလို့ရှိရင် **ctrl + d** ကို နှိပ်ပြီး ပြန်ထွက်ရပါမယ်။

နောက် ၅ မိနစ် နေတဲ့အခါ **current user's home directory** **~/** အောက်က **file1** ထဲမှာ **hello world** ဆိုတဲ့ စာသားလေး ရောက်နေပါလိမ့်မယ်။ ဒါကတော့ at ကို အသုံးပြုပြီး job တစ်ခုကို အချိန်ယေားနဲ့ run ခိုင်းတဲ့ သဘောတရားပါပဲ။

J။ ဒီလို ခိုင်းစေရာမှာ ဒီ **job** အတွက် **cpu** နဲ့ **ram** ပမာဏကို **priority** သတ်မှတ်ပေးနိုင်ပါ သေးတယ်။ **priority a** ကနေ **z** အထိ ရှိပြီး ဘာမှ မသတ်မှတ်ရင် **default priority a** နဲ့ **run** ပါတယ်။

ဥပမာ၊ စောင့်က **job** ကို **priority b** နဲ့ **run** စေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။ **at -q b now+5min**

၃။ ဒီလို အချိန်ယေားနဲ့ **run** ခိုင်းထားတဲ့ **job** တွေ ရှိမရှိ ကြည့်ချင်ရင် **atq** ဆိုတဲ့ command ကို သုံးရပါတယ်။ **at -l** လိုလည်း သုံးနိုင်ပါတယ်။ ဒီအခါ schedule ဆွဲထားတဲ့ **job** တွေကို **id** နံပါတ်နဲ့ ဖော်ပြပေးပါလိမ့်မယ်။

at -l

၄။ **job** တစ်ခု ဘယ်လို **run** မလဲ သိချင်ရင်တော့ **at** ရှေ့မှာ **-c** နဲ့ task id ကိုထည့် enter ခေါက်ပြီး ကြည့်နိုင်ပါတယ်။

ဥပမာ၊ **atq** နဲ့ ကြည့်လိုက်လို့ **job id 10** ကို တွေ့တယ်။ ဒီ **job id 10** သည် ဘယ်လို **run** မလဲ ကြည့်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

at -c 10

၅။ **job** တစ်ခု ကို ပြန်ပြီး cancel လုပ်ချင်တဲ့အခါ **atrm** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ ဥပမာ၊ **task id 10** ကို cancel လုပ်မယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command syntax ကတော့ အောက်ပါအတိုင်းပါ။

atrm 10

at ကို အသုံးချုပြီး **job** တွေကို **schedule** ဆွဲတဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။
အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(15-7-2018)



crontab ကို အသုံးပြု၍ job တစ်ခုကို အချိန်ယေားဖြင့် ပုံမှန် run စေခြင်း

လက်တွေ့လုပ်ငန်းခွင်မှာလည်း တကယ် သုံးနေရတဲ့ tool တစ်ခုပါနော်။ daily back up script တွေ run ဖို့ စသည်ဖြင့် အသုံးတည့်ပါတယ်။ linux ပေါ်မှာ **at** ဆိုတဲ့ command ကို အသုံးပြုပြီး job တစ်ခုကို run ခိုင်းတဲ့အကြောင်းကို ရေးပြီးပါပြီ။ **at** အသုံးပြုရင် job တစ်ခုကို ၁ ခါပဲ run လို့ ရပါ တယ်။ one time ပေါ့။ နေ့စဉ် လစဉ် ပုံမှန် ခိုင်းစေထားလို့ မရပါဘူး။ နေ့စဉ် လစဉ် ပုံမှန် ခိုင်းစေထားလိုတဲ့အခါ **crontab** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ သူ့ရဲ့ package name က **cronie** ဖြစ်ပြီး၊ **service name** က **crond** ဖြစ်ပြီး၊ **command** က **crontab** ပါ။ job တစ်ခုကို အချိန်ယေားနဲ့ ပုံမှန် ခိုင်းစေထားလိုတဲ့အခါ **crontab -e** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ **crontab -e** လို့ ရှိက် enter ခေါက်လိုက်ရင် **vi** နဲ့ file တစ်ခု ပွင့်လာပါလိမ့်မယ်။

အဲဒီ ပွင့်လာတဲ့ file ထဲကို အပိုင်း ၆ ပိုင်းနဲ့ အစဉ်လိုက် configure လုပ်ပေးရပါတယ်။ အဲဒီ အပိုင်း ၆ ပိုင်းကတော့ -->

၁။ **မိနစ် (0 to 59)**

၂။ **နာရီ (0 to 23)**

၃။ **day of month (1 to 31)**

၄။ **month (1 to 12)**

၅။ **day of week (0 [sunday] to 6 [saturday]) { * = ဘယ်နေ့မဆို }**

၆။ **job script (run မယ့် job)** တို့ပါပဲ။

အဲဒီ format ပုံစံကို **/etc/crontab** ဆိုတဲ့ **file** မှာ ဝင်ကြည့်နိုင်ပါတယ်။

ဥပမာ ဂျိလိုင်လ ၁၉ ရက်နေ့ မနက် ၇ နာရီ ၃၀ မိနစ် ရောက်တိုင်း current user's home directory အောက်က Desktop အောက်က aungsan.mp3 ဆိုတဲ့ သီချင်းကို ပွင့်လာစေချင်တယ် ဆိုပါစို့။

crontab -e လို့ ရှိက် enter ခေါက်ပြီး အောက်ပါအတိုင်း configure လုပ်ရပါမယ်။

i ကို နှိပ် **insert mode** နဲ့ ဝင်ပြီး အောက်ပါအတိုင်း ရေး။

30 7 19 7 * ~/Desktop/aungsan.mp3

ပြီးရင် **Esc :wq** နဲ့ ပြန်ထွက်ပေါ့။ ဒီနေရာမှာ --> 30 က မိနစ်၊ 7 က နာရီ၊ 19 က day of month၊

7 က month၊ * က ဘယ်နေ့မဆို၊ (နေ့စဉ်)၊ **~/Desktop/aungsan.mp3** က job script

ဒီလို့ အမြဲပါယ်ရပါတယ်။ နောက် job တစ်ခု ခိုင်းချင်ရင် အားဖို့ အောက်မှာပဲ နောက် ၁ကြောင်း ထပ်ရေးနိုင်ပါတယ်။ ဒါကတော့ **crontab -e** နဲ့ job ၁ ခုကို အချိန်ယေားနဲ့ ပုံမှန် run ခိုင်းတဲ့ သဘောတရားပါ။

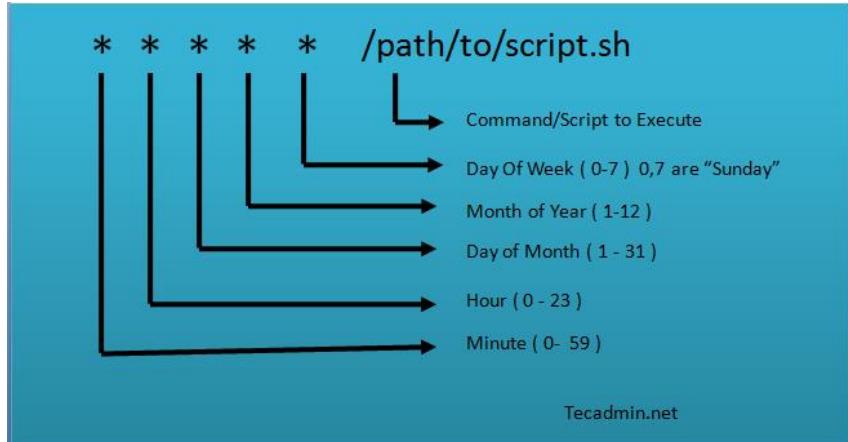
job တွေ ဘယ်နှစ်ခု ရှိလဲ သီချင်ရင် **crontab -l** နဲ့ ကြည့်နိုင်ပါတယ်။ **crontab -l**

job တွေကို ပြန်ဖျက်ချင်ရင်တော့ **crontab -r** လို့ ရှိက် **enter** ခေါက်ရုံပါပဲ။ **crontab -r**

crontab -e ကို အသုံးပြုပြီး **job** တွေကို အချင်းယော်နဲ့ ပုံမှန် ခိုင်းစေတဲ့အကြောင်းကတော့ ဒီလောက်ပါပဲ။ အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(16-7-2018)



kernel update

admin လည်း အလုပ် မပြတ်သေးလို့ ခဏ နားရင်း၊ ခုတာခါတော့ တိုတိရှင်းရှင်း ပေမယ့် လက်တွေ့လောကမှာ မကြာခဏ အသုံးပြုနေရမယ့် topic လေး ၁ ခုကို တင်လိုက်ပါတယ်နော်။ အဲဒါကတော့ linux ရဲ့ kernel ကို update လုပ်တဲ့ process ပါ။ kernel ဆိုတာ ဘာလဲ ဆိုတာကို user, application, shell, kernel and hardware ဆိုတဲ့ post မှာ တိတိကျကျ တင်ပေးခဲ့ပြီးပြီး အသေးစိတ် ထပ် မရှင်းတော့ပါဘူးနော်။ kernel ဆိုတာ linux os ရဲ့ အုတ်မြစ် core ပေါ့နော်။ linux os နဲ့ ပတ်သက်တဲ့ kernel update တွေဟာ အခါအားလုံးစွာ အသစ် ထပ်ထွက်လေ့ ရှိပါ တယ်။ အဲဒါအခါ မိမိ လက်ရှိ os ရဲ့ kernel ကို update လုပ်ပေးရတယ်ပေါ့နော်။ ၁။ kernel ကို install လုပ်ဖို့ အတွက် yum -y install kernel ဆိုတဲ့ command ကို အသုံးပြုပါတယ်။ **yum -y install kernel**

အကယ်၍ kernel ဟာ update ဖြစ်ပြီးသားဆိုရင် Nothing to do. Compete! လို့ ပြပါလိမ့်မယ်။ update လုပ်ဖို့ လိုရင်တော့ kernel အသစ်ကို သွင်းယူသွားပါလိမ့်မယ်။ ဒီသဘောပါ။ တစ်ခု ပြောချင်တာက၊ တချို့တွေက yum -y install kernel နဲ့ သွင်းပြီးသွားရင် kernel update ဖြစ်သွားပြီလို့ ထင်လေ့ရှိကြပါတယ်။ တကယ်က နောက် ဆက်လုပ်ရမယ့် အပိုင်းလေးတွေ ရှိပါ သေးတယ်။ ဆက်ရအောင်!

၂။ **yum list kernel** ဆိုတဲ့ command ကို အသုံးပြုပြီး မိမိ စက်ထဲက kernel list ကို ပြသခိုင်းပါ။ version အလိုက် ပြသပါလိမ့်မယ်။

၃။ **grub2-editenv list** လို့ ရှိက် enter ခေါက်ပါ။ လက်ရှိ kernel ကို ပြသပါလိမ့်မယ်။

၄။ **yum list kernel** က ပြသတဲ့ **list** နဲ့ **grub2-editenv** က ပြတဲ့ kernel ကို တိုက်စစ်ပြီး **version update** ဖြစ်မဖြစ် ကြည့်ပါ။

၅။ grub2-set-default "update_kernel" လို့ ရိုက်ပြီး update ဖြစ်တဲ့ kernel ကို default အဖြစ် သတ်မှတ်ပါ။ **update_kernel** နေရာမှာ နောက်ဆုံး **update** ဖြစ်တဲ့ **kernel** ၏ version ကို ထည့်ပေးရမှာပါ။ ဥပမာ >> "3.10.0-862.6.3.el7" စသည်ဖြင့်ပေါ့။

၆။ grub2-mkconfig -o /boot/grub2/grub.cfg လို့ ရိုက် enter ခေါက်ပြီး။ kernel ကို refresh လုပ်ပေးပါ။

၇။ init 6 လို့ ရိုက် enter ခေါက်ပြီး စက်ကို restart လုပ်ပါ။ ဒါမှ kernel အသက်ဝင်ပါလိမ့်မယ်။ အဲအဆင့်တွေ ပြီးရင်တော့ kernel update လုပ်ငန်း ပြီးဆုံးပြီပေါ့။ အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(17-7-2018)

```

root@localhost ~# yum list kernel
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: centos.mirror.iweb.ca
* extras: centos.mirror.iweb.ca
* updates: centos.mirror.iweb.ca
Installed Packages
kernel.x86_64                  3.10.0-327.el7          @anaconda
kernel.x86_64                  3.10.0-862.3.3.el7      @updates
kernel.x86_64                  3.10.0-862.6.3.el7      @updates
root@localhost ~# grub2-editenv list
saved_entry=3.10.0-862.6.3.el7
root@localhost ~# grub2-set-default "3.10.0-862.6.3.el7"
root@localhost ~# grub2-mkconfig -o /boot/grub2/grub.cfg

```

system cron jobs

ဒါလေးက အရေးမကြိုးပေမယ့် သိထားတော့ ကောင်းတာပေါ့။ **at** တို့ **crontab** တို့ စတဲ့ command တွေကို အသုံးပြုပြီး job တွေကို အချိန်ယေားနဲ့ စေခိုင်းတဲ့အကြောင်းကတော့ ရေးပြီးခဲ့ပါပြီ။ ခုတစ်ခါ တင်ဆက်မှာက system အနေနဲ့ တချို့ **job** တွေကို သူ့ဖို့သာ အချိန်ယေားနဲ့ ဘယ်လို **run** နေသလဲဆိုတာပါပဲ။ သဘောတရားလေးပေါ့။

၁။ **system cron job** နဲ့ ဆိုင်တဲ့ file တွေဟာ **/etc/cron.d** အောက်မှာ ရှိပါတယ်။

၂။ **system** အနေနဲ့ **job** တွေကို **run** တဲ့အခါ **run-parts** ဆိုတဲ့ program က နောက်ကွယ်ကနေ အလုပ်လုပ်ပေးပါတယ်။ **run** တဲ့ ပုံစံကို **/etc/anacrontab** ထဲမှာ ဖော်ပြထားပါတယ်။ **vi** နဲ့ ဝင်ကြည့်နိုင်ပါတယ်။ **vi /etc/anacrontab**

၃။ နေ့ရက်အလိုက် **run** ရမယ့် **job** တွေကို **/etc/cron.daily** ဆိုတဲ့ directory အောက်မှာ သွား **run** ပါတယ်။ ဝင်ပြီး လေ့လာနိုင်ပါတယ်နော်။ **vi /etc/cron.daily**

၄။ အပတ်စဉ် **run** ရမယ့် **job** တွေကို **/etc/cron.weekly** ဆိုတဲ့ directory အောက်မှာ သွား **run** ပါတယ်။ **vi /etc/cron.weekly**

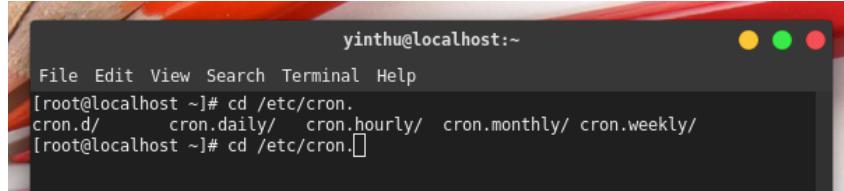
၆။ လစဉ် **run** ရမယ့် **job** တွေကို **/etc/cron.monthly** ဆိုတဲ့ directory အောက်မှာ သွား **run** ပါတယ်။ **vi /etc/cron.monthly**

၃။ နာရီ အလိုက် **run** ရမယ့် **job** တွေကိုတော့ **/etc/cron.d/0hourly** ဆိုတဲ့ **file** က **/etc/cron.hourly** ဆိုတဲ့ **directory** အောက်မှာ သွား **run** ပါတယ်။ **/etc/cron.d/0hourly** ဆိုတဲ့ **file** ကို **vi text editor** နဲ့ဖွင့်ကြည့်ရင် မြင်နိုင်ပါတယ်။ **vi /etc/cron.d/0hourly system** အနေနဲ့ **job** တွေကို အချိန်ယေားနဲ့ **run** တဲ့ သဘောတရားကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(22-7-2018)



SELinux (Security Enhanced Linux)

linux မှာ SELinux ဆိုတာ **firewall** မဟုတ်ပဲ **security** အတွက် ထားထားတဲ့ **layer** တစ်ခုပါ။ ဥပမာ **http** ကို ဖွင့်ပေးထားရင်တောင်မှ SELinux က ခွင့်ပြုထားတဲ့ **file** တွေကိုပဲ **access** လုပ်လို့ ရမယ်။ ဒီသဘောပါ။ SELinux ရဲ့အရှည်က Security Enhance Linux ပါ။ SELinux ကို ကောင်း ကောင်း အသုံးချိန်ဖို့အတွက် **policycoreutils-python** ဆိုတဲ့ **package** နဲ့ **setroubleshoot-server** ဆိုတဲ့ **package** လိုအပ်ပါတယ်။ မရှိရင် **yum install** နဲ့ သွင်းပေးရပါမယ်။ ကဲ ! လေ့လာကြည့်ရအောင်။

အရင်ဆုံး **root account** နဲ့ **login** ဝင်ပြီး **mkdir dir1** ဆိုတဲ့ **directory** လေး ၁ ခု **touch file1** နဲ့ **file** လေး ၁ ခု ဆောက်လိုက်ပါ။ **file** နဲ့ **directory** တွေမှာ SELinux က သတ်မှတ်ပေးထားတဲ့ **label** လေးတွေ ရှိပါတယ်။ ဒါကို **context** လို့ ခေါ်ပါတယ်။ ဒီ **context** လေးတွေပေါ်မှုတည်ပြီး **access** လုပ်ခွင့်တွေ ပြောင်းလဲသွားတာပါ။ **file** မှာ ပါတဲ့ **context** ကို ကြည့်ချင်ရင် **ls -lZ** ကို အသုံးပြုရပါတယ်။ **ls -lZ file1**

directory မှာ ပါတဲ့ **context** ကို ကြည့်ချင်ရင် **ls -lZ** ကို အသုံးပြုရပါတယ်။ **ls -lZ dir1** **process** တွေမှာ ပါတဲ့ **context** တွေကို ကြည့်ချင်ရင် **ps -auxZ** နဲ့ ကြည့်ရပါတယ်။ **ps -auxZ** ဟုတ်ပြီ။ ဆက်သွားကြရအောင်။ SELinux မှာ **mode** ၃ မျိုး ရှိပါတယ်။ ဒါတွေကတော့၊

1. **enforcing**

2. **permissive**

3. **disabled** တို့ပါပဲ။

1. **enforcing** က **rule** နဲ့ မညီတာတွေကို ပိတ်ပစ်ပါတယ်။

2. **permissive** က **rule** နဲ့ မညီတာတွေကို ပိတ်မပစ်ပါဘူး။ ဒါပေမယ့် ဘာတွေဖြစ်သွားတယ် ဆိုတာကို **alert** လုပ်ပါတယ်။

3. **disabled** ကတေသာ **SELinux** ကို လုံးဝ ပိတ်ထားတာပါ။

SELinux သည် ဘယ် **mode** မှာ ရှိနေသလဲ ကြည့်ချင်ရင် **getenforce** ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ **getenforce**

SELinux ရဲ့ **mode** ကို ပြောင်းချင်ရင် **setenforce** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။

1. **enforcing** ကို ပြောင်းချင်ရင် **setenforce 1**

2. **permissive** ကို ပြောင်းချင်ရင် **setenforce 0**

3. **disabled** လုပ်ချင်ရင် **/etc/selinux/config** ကို **vi** နဲ့ ဝင်ပြီး **SELINUX=disabled** လို့ ပြင်ပေးရပါမယ်။

ကဲ ! ဆက်လိုက်ရအောင်။

file နဲ့ **directory** တွေ အတွက် **context** တွေကို ယာယို (**temporary**) နဲ့ အမြဲ (**permanent**) ဆိုပြီး သတ်မှတ်ပေးနိုင်ပါတယ်။

1. **temporary** သတ်မှတ်ချင်ရင် **chcon -t** ကို အသုံးပြုရပါတယ်။ **reboot** လုပ်လိုက်ရင် ပြန်ပျက်ပါတယ်။ ဥပမာ **chcon -t httpd_sys_content_t file1**

2. **permanent** သတ်မှတ်ချင်ရင် **semanage fcontext -a -t** ကို အသုံးပြုရပါတယ်။ **reboot** လုပ်လည်း မပျက်တော့ပါဘူး။ **semanage fcontext -a -t "httpd_sys_content_t" dir1**

3. **context** တစ်ခုကို **permanent** သတ်မှတ်ပြီးရင် **update** ဖြစ်သွားအောင် **restorecon -RFv** ကို run ပေးရပါတယ်။ **restorecon -RFv dir1**

directory နဲ့ သူ့အောက်က **sub dir** တွေ **file** တွေကိုပါ တစ်ခါတည်း သတ်မှတ်ပေးချင်ရင် **(/.*)?**

လေးထည့်ပေးရပါတယ်။ **semanage fcontext -a -t "httpd_sys_content_t" "dir1(/.*)?"**

ဒီနေရာမှာ **httpd_sys_content_t** သည် **context** ပါ။ ဒီလို သတ်မှတ်ခြင်းအားဖြင့် **http** ကနေ တစ်ဆင့် ဒီ **directory** ကို **access** လုပ်ခွင့် ရသွားပါတယ်။

ကဲ ! ထပ်ဆက်လိုက်ရအောင်။

SELinux ရဲ့ပေးထားတဲ့ **policy** တွေကို **on/off** ဖွင့်ပိတ်လုပ်ထားနိုင်ပါတယ်။

1. **policy** တွေကို ကြည့်ချင်ရင် **getsebool -a** နဲ့ ကြည့်နိုင်ပါတယ်။ **policy** ပေါင်း **JRE** ရှိပါတယ်။ **getsebool -a**

အကြောင်းအရာတွေနဲ့ ပြည့်ပြည့်စုစု ကြည့်ချင်ရင် **semanage boolean -l** ကို အသုံးပြုရပါတယ်။

semanage boolean -l

2. **policy** တွေကို ပြင်မယ်ဆို **setsebool** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ ယာယို ပြင်မယ်ဆို **setsebool** ကို သုံးရုံပါပဲ။ **setsebool httpd_enable_homedirs on** ယာယိုသတ်မှတ်ထားတဲ့ **policy** တွေကို ကြည့်ချင်ရင် **setsebool -a** ကို အသုံးပြုရပါတယ်။ **setsebool -a**

3. **policy** ကို **permanent** သတ်မှတ်ချင်ရင် **-p** ဆိုတဲ့ **option** ထည့်ပေးရပါတယ်။

setsebool -p httpd_enable_homedirs on

ကဲ ! trouble shoot အကြောင်းလေး ဆက်လိုက်ရအောင်။

file တစ်ခုခု **directory** တစ်ခုခုကို **access** လုပ်လို့ မရရင် **error** တက်တတ်ပါတယ်။ ဒါဟာ **SELinux** က ပိုတ်ထားလိုပါ။

ဒီအခါ **system log** ကို **tail -f -n 30** နဲ့ ဝင်ဖတ်ပြီး **sealert** ဆိုတဲ့ **code** ကို ရှာပြီး **run** ပေးရပါတယ်။

ဒီအခါ **SELinux** နဲ့ ဘယ်လို့ ပြင်ဆင်ပေးဖို့ လိုအပ်နေသလဲဆိုတာ မြင်တွေ့ရပါလိမ့်မယ်။

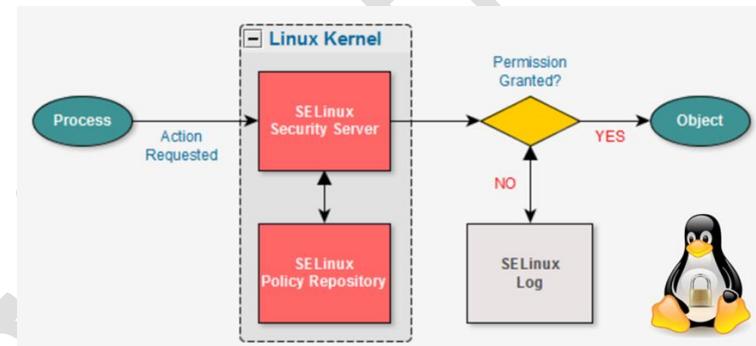
tail -f -n 30 /var/log/messages

ကဲ ! **SELinux** ရဲ့ အခြေခံ သဘောတရားတွေကတော့ ဒီလောက်ပါပဲ။ နည်းနည်း ရှုပ်ရင် ခုလောက် ပြန်ဖတ်ကြည့်ပါ လို့။ google လေးနဲ့ တွဲပြီး လွှဲလာပါလို့။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(23-7-2018)



HD (hard disk) အား partition ပိုင်းခြင်း format ချခြင်း mount လုပ်ခြင်း

ဒီတစ်ခါတွေ့ linux မှာ HD ကို

၁။ partition ပိုင်းတာ

၂။ ရလာတဲ့ partition ကို format ချတာ

၃။ format ချထားတဲ့ partition ကို မိမိ သတ်မှတ်ထားတဲ့ directory မှာ mount လုပ် တပ်ဆင်တာ စသဖြင့် လွှဲလာကြည့်ရအောင်။

lsblk လို့ ရိုက် enter ခေါက်လိုက်ရင် HD တွေရဲ့ စာရင်းကို တွေ့ရမှာပါ။ HD က -->

၁။ **IDE** အမျိုးအစားဆိုရင် **hda, hdb**

၂။ **SATA** အမျိုးအစားဆိုရင် **sda, sdb**

၃။ **virtual HD** ဆိုရင် **vda, vdb** စသဖြင့် မြင်ရပါလိမ့်မယ်။ ဒီနေရာမှာ နမူနာအဖြစ် **HD** ကို **sdb** လို့ မှတ်ထားပေးပါ။

ကဲ ! **sdb** ဆိုတဲ့ **HD** ကို **partition** ပိုင်းကြည့်ရအောင်။

၁။ **fdisk /dev/sdb** လို့ ရှိက် enter ခေါက်လိုက်ပါ။

၂။ **Command (m for help)** : လို့ ပေါ်လာပါလိမ့်မယ်။ **partition** အသစ် ပိုင်းမှာဖြစ်တဲ့အတွက် n လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၃။ **Partition type** ကို ရွှေးချိုင်းပါလိမ့်မယ်။ **primary partition** အနေနဲ့ ပိုင်းမှာဖြစ်လို့ p လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၄။ **Partition number** ဖြည့်ခိုင်းပါလိမ့်မယ်။ ပထမဆုံး **partition** ဖြစ်တောက်ခြား 1 လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၅။ **First sector** ထည့်ခိုင်းပါလိမ့်မယ်။ **blank** အတိုင်းထားပြီး enter ခေါက်ပေးလိုက်ပါ။

၆။ **Last sector** ထည့်ခိုင်းပါလိမ့်မယ်။ ဒီဥပမာမှာ 2G ပိုင်းမှာ ဖြစ်တဲ့ အတွက် +2G လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၇။ **Command (m for help)** : ဆိုတဲ့နေရာကို ပြန်ရောက်သွားပါလိမ့်မယ်။ **partition ID** သတ်မှတ်ပေးဖို့ အတွက် t လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၈။ **Hex code (type L to list all codes)** : လို့ တွေ့ရပါမယ်။ L လို့ ရှိက် enter ခေါက်ပေးလိုက်တဲ့အခါ **code** တွေကိုတွေ့ရပါမယ်။ **linux** အတွက် **ID** က 83, **swap** အတွက် **ID** က 82, **LVM** အတွက် **ID** က 8e ပါ။ ဒီနေရာမှာ 83 လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၉။ **Command (m for help)** : ဆီ ပြန်ရောက်လာပါလိမ့်မယ်။ w လို့ ရှိက် enter ခေါက်ပြီး save လိုက်ပါ။

၁၀။ ဒါဆို **partition** တစ်ခု ဖန်တီးပြီးသွားပါပြီ။ **partition** အသစ်ကို **system** က သိအောင် **partprobe** လို့ ရှိက် enter ခေါက်ပေးလိုက်ပါ။

၁၁။ **lsblk** ဆိုတဲ့ **command** ကို သုံးပြီး ပြန်ကြည့်လိုက်ပါ။ **/dev/sdb1** ဆိုတဲ့ **partition** တစ်ခု တိုးလာတာကို မြင်ရပါလိမ့်မယ်။ **partition** ပိုင်းတာကတော့ ဒီလောက်ပါပဲ။

Format ချုတဲ့အကြောင်း ဆက်သွားကြရအောင်။

Format ချို့ အတွက် **mkfs** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။

၁။ **xfs** နဲ့ **format** ချုမယ်ဆို **mkfs.xfs**

၂။ **ext4** နဲ့ **format** ချုမယ်ဆို **mkfs.ext4** ဒီလို အသုံးပြုရပါတယ်။

စောောက ပိုင်းထားတဲ့ **/dev/sdb1** ကို **ext4** နဲ့ **format** ချုမယ်ဆိုပါစို့။ အသုံးပြုရမယ့် **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။ **mkfs.ext4 /dev/sdb1**

ဒါကတော့ **format** ချုတဲ့အကြောင်းပါပဲ။ **format** ချိုပြီးတဲ့နောက် အသုံးပြုဖို့ **file** တွေ ထည့်နိုင်ဖို့ **directory** တစ်ခုခုမှာ **mount** လုပ်တပ်ဆင်ရပါတယ်။

သင်ဟာ **root account** နဲ့ **login** ဝင်ထားတယ် ဆိုပါစို့။ / **directory** အောက်မှာ **mkdir hd1** လို့ **directory** လေး ၁ ခု ဆောက်လိုက်ပါ။ **format** ချုပြုးသား **partition** ကို **hd1** ဆိုတဲ့ **directory** ဆီ **mount** လုပ်မယ်ဆိုပါစို့။ အသုံးပြုရမယ့် **command syntax** ကတော့ အောက်ပါအတိုင်းပါ။
mount /dev/sdb1 /hd1

ဒါက ယာယို **mount** လုပ်တာပါ။ **reboot** လုပ်လိုက်ရင် ပြန်ပြတ်ပါတယ်။ အမြဲ mount လုပ်ထား ချင်ရင် **/etc/fstab** ကို **vi** နဲ့ ဖွင့်ပြီး အောက်ပါအတိုင်း ရေးသွင်းပေးရပါမယ်။ **vi /etc/fstab**

၁။ **vi /etc/fstab** ။ ၂ ကို နိုင် **insert mode** ကို ဝင်

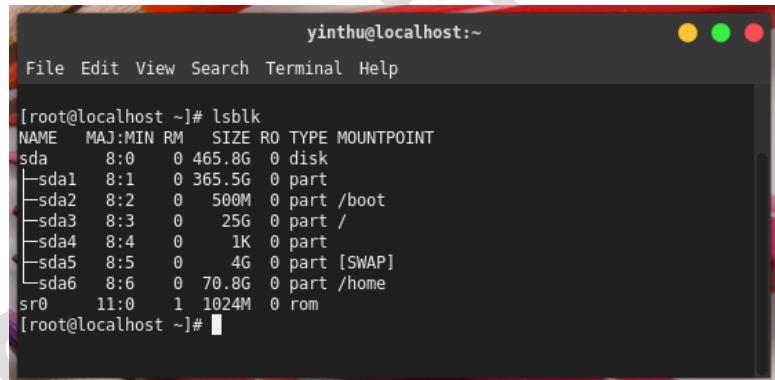
၃။ **/dev/sdb1 /hd1 ext4 defaults 0 0** လို့ ရေး

၄။ Esc နိုင် **:wq** နဲ့ ပြန်ထွက် ဒါဆို အမြဲ **mount** ဖြစ်သွားပါပြီ။ **cd /hd1** ကို ဝင်ပြီး file တွေ သိမ်းဆည်းနိုင်ပါပြီ။ ကဲ ! HD ကို partition ပိုင်း၊ format ချုံ directory မှာ mount လုပ် တပ်ဆင်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(30-7-2018)



```
yinthu@localhost:~$ lsblk
[...]
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda    8:0    0 465.8G  0 disk 
|---sda1 8:1    0 365.5G  0 part /boot
|---sda2 8:2    0 500M  0 part /boot
|---sda3 8:3    0 25G   0 part /
|---sda4 8:4    0 1K    0 part
|---sda5 8:5    0 4G    0 part [SWAP]
└---sda6 8:6    0 70.8G  0 part /home
sr0   11:0   1 1024M 0 rom 

yinthu@localhost:~$
```

virtual memory (swap)

linux မှာ **Hard Disk (HD)** ကို **partition** ပိုင်းတာ **format** ချုတာ၊ **directory** မှာ **mount** လုပ်တာတွေကို တင်ခဲ့ပြီးပါပြီ။

ဒီတစ်ခါ **virtual memory (RAM)** အတူ) ဖန်တီးတဲ့အကြောင်းကို လေ့လာကြည့်ရအောင်။

physical ram မလောက်တဲ့အခါ **virtual memory** ဟာ **ram** လိုပဲ အလုပ်လုပ်ပေးပါတယ်။

linux မှာ **virtual memory** ကို **swap** လို့ ခေါ်ပါတယ်။ **free -h** ဆိုတဲ့ command နဲ့ ဝင်ကြည့်မယ် ဆိုရင် **swap** ကို တွေ့ရပါလိမ့်မယ်။ **free -h**

ကဲ ! **swap** တစ်ခုကို ဘယ်လို ဖန်တီးမလဲ စတင်လိုက်ရအောင်။

၁။ **HD** ကို **partition** ပိုင်းကတည်းက **type ID** ကို **82** လို့ သတ်မှတ်ပေးလိုက်ပါ။ ဒီနေရာမှာ နမူနာအဖြစ် **partition** ကို **/dev/sdb1** လို့ ဆိုကြပါစို့။ **size** က **1G** ထားပါစို့။

JII **sdb1** ရလာပြီးတဲ့နောက် သူ့ကို **format** ချုပါမယ်။ **storage** အနေနဲ့ မဟုတ်ပဲ **virtual memory** ဖန်တီးမှာ ဖြစ်လို့ **mkswap** ဆိုတဲ့ command ကို အသုံးပြုရပါမယ်။
mkswap /dev/sdb1 လို့ ရှိက် **enter** ခေါက်လိုက်ပါ။ ဒါဆို **swap** တစ်ခု ရပါပြီ။

၃။ **swap** ကို **virtual memory** အဖြစ် တပ်ဆင်ဖို့ **swapon** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ **swapon /dev/sdb1** လို့ ရှိက် **enter** ခေါက်လိုက်ပါ။

၄။ **free -h** လို့ ရှိက် **enter** ခေါက်လိုက်ပါ။ **ram** နေရာမှာ **swap** အနေနဲ့ **1G** ထက်တိုးလာတာ တွေ့ရပါလိမ့်မယ်။

၅။ ဒါက **swap** ကို ယာယို **mount** တာပါ။ စက်ကို **restart** ပေးလိုက်ပြီး ပြန်တက်လာရင် ရှိမှာ မဟုတ်တော့ပါဘူး။

အဲတော့ အမြဲ **mount** ထားဖို့ အတွက် **/etc/fstab** ထဲမှာ အောက်ပါအတိုင်း ဝင်ရေးပေးရပါမယ်နော်။ **/dev/sdb1 swap swap defaults 0 0**

၆။ **virtual memory** အဖြစ် တပ်ဆင်ထားတဲ့ **swap** ကို ပြန်ဖြေတွက်ချင်တယ် ဆိုပါစွဲ။ **swapoff** ဆိုတဲ့ **command** ကို အသုံးပြုရပါတယ်။ **swapoff /dev/sdb1** လို့ ရှိက် **enter** ခေါက်လိုက်ပါ။

free -h နဲ့ ပြန်ကြည့်ပါ။ **swap** ရဲ့ **size** ပြန်လေ့သွားတာကို တွေ့ရပါလိမ့်မယ်။ အသုံးပြုခဲ့တဲ့ **command** တွေကို ပြန်ကြည့်မယ်ဆိုရင် -->

၇။ **HD** ကို **partition** ပိုင်းစပ်မှာ **type ID** ကို **82** လို့ ပေး

JII **mkswap /dev/sdb1** Virtual Memory ကိုဖန်တီးတာ

၈။ **swapon /dev/sdb1** Swap ကို Virtual Memory အဖြစ်တပ်ဆင်တာ

၉။ **free -h** swap ရှိမရှိ ပြန်စစ်တာ

၁၀။ အမြဲ **mount** ထားဖို့ **/etc/fstab** ထဲမှာ ဝင်ရေး

၆။ ပြန်ဖြေမယ်ဆို **swapoff /dev/sdb1** လို့ တွေ့ရမှာပါ။

linux မှာ **swap** တစ်ခု ဖန်တီးပြီး **virtual memory** အဖြစ် တပ်ဆင် အသုံးပြုတဲ့အကြောင်း ကတော့ ဒီလောက်ပါပဲ။ စာဖတ်ရင်း တစ်ဆင့်ချင်း စမ်းလှပ်သွားရင် အဆင်ပြေပါလိမ့်မယ်။ **swap** နဲ့ ပတ်သက်တာတွေ မပြုလုပ်ခင် ယခင် **post** မှာ တင်ထားတဲ့ **HD** ကို **partition** ပိုင်းတာ၊ **type ID** သတ်မှတ်တာတွေကို နားလည်ထားဖို့တော့လိုပါတယ်။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(31-7-2018)



```

centos 7-2 [Running] - Oracle VM VirtualBox
[root@centos ~]# mkswap /dev/sdb1
Setting up swapspace version 1, size = 1048572 KiB
no label, UUID=a2dc97a4-db7d-454a-8c5d-13227ce663f3
[root@centos ~]# free -h
Mem:      991M     64M    832M   6.7M    95M   801M
Swap:        819M      0B    819M
[root@centos ~]# swapon /dev/sdb1
[root@centos ~]# free -h
Mem:      991M     64M    831M   6.7M    95M   800M
[root@centos ~]#

```

+ + [linux firewall (iptables)] + +

What is iptables?

iptables ဆိုတာ **linux** ရဲ့ **firewall** တစ်ခုပါ။ **firewall** မှာ **client firewall** နဲ့ **network firewall** ဆိုပြီး၂ မျိုး ရှိပါတယ်။ ဒီနေရာမှာ **iptables** ကို **client firewall** အနေနဲ့ အသုံးပြုပုံကို ပြောပြသွားမှာပါ။

firewall/iptables ဘယ်လို အလုပ်လုပ်သလဲ?

iptables ဟာ **network traffic** (1. **inbound**, 2. **outbound**) တွေကို **filter** ခံပြီး **policy** တွေနဲ့ ထိန်းချုပ်ပေးပါတယ်။

What is network traffic?

network ပေါ်မှာ သွားနေတဲ့ အရာမှုနှင့်သမျှကို **network traffic** လို့ ခေါ်ပါတယ်။ 1-**inbound** နဲ့ 2-**outbound traffic** ဆိုပြီး၂ မျိုး ရှိပါတယ်။

What is inbound traffic?

မိမိရဲ့ **ip** ဆိုကို ဝင်လာတဲ့ **traffic** တွေကို **inbound traffic** လို့ ခေါ်ပါတယ်။

what is outbound traffic?

မိမိရဲ့ **ip** ကနေ အခြား **network** ဆီ ထွက်သွားတဲ့ **traffic** တွေကို **outbound traffic** လို့ ခေါ်ပါတယ်။

What is firewall rule/policy?

အဲဒီ **traffic** တွေကို ထိန်းချုပ်ဖို့ အတွက် **firewall rules/policies** တွေ ဖန်တီးပေးရပြီး၊ အဲဒီ **rules** ၏ ခုံမှာ အောက်ပါ **parameter** တွေ ပါဝင်ရပါတယ်။

၁။ **source ip**

၂။ **destination ip**

၃။ **protocal** အမျိုးအစား

၄။ **ACCEPT**(လက်ခံ) သို့မဟုတ် **DROP**(ဖြတ်တောက်) တို့ ဖြစ်ပါတယ်။

iptables ရဲ့ **command** တွေကတော့ ...

1. **iptables** ရဲ့ **inbound policy** ကို ကြည့်ဖို့အတွက် **iptables -L INPUT** ကို သုံးပါတယ်။
2. **iptables** ရဲ့ **outbound policy** ကို ကြည့်ဖို့အတွက် **iptables -L OUTPUT** ကို သုံးပါတယ်။
3. **iptables** ရဲ့ **policy** တွေကို **save** ဖို့ အတွက် **service iptables save** ကို သုံးပါတယ်။
4. **iptables** ရဲ့ **policy** တွေကို အကုန် **delete** လုပ်ဖို့ အတွက် **iptables -F INPUT** နဲ့ **iptables -F OUTPUT** ကို သုံးပါတယ်။
5. **iptables** ရဲ့ **policy** တစ်ခုချင်းကို **delete** လုပ်ဖို့အတွက် **iptables -D** ကို သုံးပါတယ်။

6. **iptables** ကို **restart** လုပ်ဖို့အတွက် **service iptables restart** ကို သုံးပါတယ်။
7. **iptables** ကို **operating system** နဲ့အတူ စတက်လာဖို့အတွက် **chkconfig iptables on** ကို သုံးပါတယ်။
8. **iptables** သုံးပုံသုံးနည်းကို **iptables --help** နဲ့ ကြည့်နိုင်ပါတယ်။
ကဲ ! **rule/policy** တွေ သတ်မှတ် ကြည့်ရအောင်
မိမိစက်ရဲ **ip** က **192.168.1.1** ဖြစ်တယ် ဆိုပါစို့။ တခြားစက်ရဲ **ip** က **192.168.1.2** ဆိုပါစို့။
၁။ **example** အနေနဲ့ မိမိစက်ကို **192.168.1.2** ကနေ **ping** မရအောင် **rule** ကို အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါတယ်။ **iptables -A INPUT -s 192.168.1.2 -d 192.168.1.1 -p ICMP -j DROP**
ဒီ **rule** ကို ရှင်းပြရရင် -

-A သည် **append** ရေးသွင်းမယ်။

INPUT ဆိုတာက မိမိဆီ တခြားက လာမှာဖြစ်လို့ **inbound rule** လို့ အဓိပ္ပာယ်ရပါတယ်။

-s သည် **source ip** ကို ကိုယ်စားပြုပါတယ်။

-d သည် **destination ip** ကို ကိုယ်စားပြုပါတယ်။

-p သည် **protocol** ကို ကိုယ်စားပြုပြီး သူ့နောက်မှ **protocol** အမျိုးအစား ထည့်ပေးပါတယ်။
ဒီနေရာမှာ **ping** ဖြစ်တဲ့အတွက် **protocol** အမျိုးအစားက **ICMP** ပါ။

-j က **jump** လို့ အဓိပ္ပာယ်ရပါတယ်။

DROP ကတော့ ပယ်ချမယ်လို့ အဓိပ္ပာယ်ရပါတယ်။ လက်ခံမယ်ဆို **ACCEPT** ပါ။

ဒီလို့ သတ်မှတ်လိုက်ရင် မိမိစက် **destination ip 192.168.1.1** ကို **source ip 192.168.1.2** ကနေ **ping** လို့ မရတော့ပါဘူး။

၂။ **example** အနေနဲ့ မိမိစက်ကို **192.168.1.2** ကနေ **ssh** နဲ့ ဝင်လို့ မရအောင် **rule** ကို အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါတယ်။

iptables -A INPUT -s 192.168.1.2 -d 192.168.1.1 -p TCP --dport 22 -j DROP

ဒီ **rule** ကို ရှင်းပြရရင် -

-A သည် **append** ရေးသွင်းမယ်။

INPUT ဆိုတာက မိမိဆီ တခြားက လာမှာဖြစ်လို့ **inbound rule** လို့ အဓိပ္ပာယ်ရပါတယ်။

-s သည် **source ip** ကို ကိုယ်စားပြုပါတယ်။

-d သည် **destination ip** ကို ကိုယ်စားပြုပါတယ်။

-p သည် **protocol** ကို ကိုယ်စားပြုပြီး သူ့နောက်မှ **protocol** အမျိုးအစား ထည့်ပေးပါတယ်။
ဒီနေရာမှာ **ssh** ဖြစ်တဲ့အတွက် **protocol** အမျိုးအစားက **TCPP** ပါ။

--dport က ဝင်လာမယ့် **port** နံပါတ်ပါ။ ဒီနေရာမှာ **ssh** မို့ **port no.** က 22 ပါ။

-j က **jump** လို့ အဓိပ္ပာယ်ရပါတယ်။

DROP ကတော့ ပယ်ချမယ်လို့ အဓိပ္ပာယ်ရပါတယ်။ လက်ခံမယ်ဆို **ACCEPT** ပါ။

ဒီလို့ သတ်မှတ်လိုက်ရင် မိမိစက် **destination ip 192.168.1.1** ကို **source ip 192.168.1.2**

ကနေ **ssh** နဲ့ ဝင်လို့ မရတော့ပါဘူး။ ခုလောက်ဆို သဘောတရားကို နဲ့နားလည်လောက်ပြီ ထင်ပါတယ်။ network ထဲမှာ စက် ၃ လုံး ရှိတယ်။

၁။ မိမိစက်ရဲ့ **ip ၏ 192.168.1.1**

၂။ တရုတ်မရဲ့စက် **ip ၏ 192.168.1.2**

၃။ ဗုံးမရဲ့စက် **ip ၏ 192.168.1.3** ဆိုပါစို့။

မိမိစက်ကို တရုတ်မရဲ့စက်ကနေ **ping** လို့ ရပါး၊ ဗုံးမရဲ့စက်ကနေ **ping** လို့ မရအောင် **iptables** နဲ့ **rule** သတ်မှတ်ကြည့်ကြဖို့။

iptables -A INPUT -s 192.168.1.2 -d 192.168.1.1 -p ICMP -j ACCEPT

iptabes -A INPUT -s 192.168.1.3 -d 192.168.1.1 -p ICMP -j DROP

service iptables save

ကဲ ! သဘောတရားကတော့ ဒီလောက်ပါပဲ။ တရုတ်မရဲ့ စက်ကနေ **ssh** နဲ့ ဝင်လို့မရပဲ၊ ဗုံးမရဲ့စက်ကနေ **ssh** နဲ့ ဝင်လို့ရအောင် rule ကွဲကိုယ် သတ်မှတ်ကြည့်ပါ။

သတိပြုရန် >> rule တွေ ထည့်သွင်းရာမှာ **iptables** ဟာ ပထမ no.1 rule ကို ဆောင်ရွက်ပါးမှ no.2 rule ကို ဆက်လက် ဆောင်ရွက်ပါတယ်။

လသမျှ ping အားလုံးကို ပိတ်ထားတာကို အရင်ထည့်ပါးမှ 192.168.1.2 အတွက် ping ဖွင့်ပေးတဲ့ rule ကို ထည့်ရင်၊ အရာရောက်တော့မည် မဟုတ်ပါ။

192.168.1.2 ကို ping ဖွင့်ပေးတာကို အရင်ထည့်ပါးမှ ကျွန်တာတွေကို ping ပိတ်တာကို ဒုတိယနေရာက ထည့်ရပါမယ်။

iptables အနေနဲ့ **network firewall** အဖြစ် **postrouting**, **prerouting** တွေနဲ့ သုံးလို့ရတာတွေ အများကြီး ရှိပါသေးတယ်။ google, youtube ကနေ ဆက်လက် လွှဲလာကြပါလို့။ ဒီနေရာမှာ ကတော့ concept သဘောတရား အမြဲ့မြဲမျှသာပါပဲ။

+ + [linux firewall (iptables) ရဲ့ rules များကို ကြည့်ခြင်း] + +

iptables မှာ **rules** တွေကို ဘယ်လို့ ပြန်ကြည့်သလဲဆိုတာ ပြောကြပို့။ လွယ်လွယ်လေးပါ။

iptables ရဲ့ **rules** တွေကို ကြည့်ဖို့ **terminal** မှာ **iptables** ဆိုတဲ့ command နဲ့အတူ **-nL** ဆိုတဲ့ option ကို သုံးရပါတယ်။

line no. တွေနဲ့ ကြည့်ချင်ရင် **--line-numbers** ဆိုတဲ့ option ကို ထပ်ထည့်ပေးရပါတယ်။

အပြင်ကနေအတွင်းကိုလာတဲ့ traffic ကို ကာတာက input rules။ အတွင်းကနေအပြင်ကိုသွားတဲ့ traffic ကို ကာတာက output rules။ ဒီလိုမှတ်ပါ။

#**input policy** ကို ကြည့်ဖို့ --> iptables -nL INPUT --line-numbers

#**output policy** ကို ကြည့်ဖို့ --> iptables -nL OUTPUT --line-numbers

#**policy** အားလုံးကို ကြည့်ဖို့ --> iptables -nL --line-numbers ကဲ ! ဒီလောက်ပါပဲ။

+ + [inserting new rules to iptables (firewall)] + +

ခုတခါ linux ရဲ့ firewall ဖြစ်တဲ့ iptables မှာ ရှိပါသေး rules တွေ ကြားထဲကို rules အသစ်တွေ ထပ်ထည့်ချင်တဲ့အခါ ဘယ်လို လုပ်ရမလဲ ပြောကြည့်ကြဖို့။
ဥပမာ ပေးရရင် ဒီလိုပျော်ရွောများ --->

iptables မှာ rules တွေ သတ်မှတ်လိုက်တယ်။ ၁၀ ကြောင်းဆိုပါတော့။ ထည့်ပိုး save ပိုးမှ ပြန်စစ်ကြည့်တဲ့အခါ ၅ ကြောင်းမြောက်နေရာမှာ rule ၁ ခု ထပ်ထည့်ဖို့ လိုနေတာ သတိပြုမိသွားတယ် ဆိုပါတော့ပျော်ရွောများ။ ခုပြောမှာက အဲဒီအဖြစ်မျိုး ကြိုတဲ့အခါ ဘယ်လို လုပ်ရမယ်ဆိုတဲ့အကြောင်းပါ။ rule အသစ် ၁ ခုကို ထပ်ထည့်ချင်တဲ့အခါ လုပ်ရမယ့် အဆင့်တွေကတော့ အောက်ပါအတိုင်းပါပဲ။၁။ iptables -nL --line-numbers ဆိုတဲ့ command ကို သုံးပါ။ ဒီအခါ iptables မှာ မိမိ ထည့်သွေးတဲ့ rules တွေကို နံပါတ်စဉ်နဲ့ မြင်ရပါမယ်။

၂။ rule အသစ်ကို ဘယ်နှစ်ကြောင်းမြောက်၊ ဘယ်နံပါတ်စဉ်မှာ ထည့်ချင်တာလဲ၊ မှတ်သားပါ။

၃။ ပီးရင် -I ဆိုတဲ့ option ကို သုံးပါ။ ဘယ်လို သုံးရမလဲဆိုတော့ ...

ဥပမာ >> iptables ရဲ့ inbound policy မှာ 192.168.1.11 ကနေ မိမိစက်ရဲ့ ip 192.168.1.10 ဆီ ssh နဲ့ ဝင်လို့မရအောင် ကာတဲ့ rule ၁ ခု ရေးမယ်။ အဲဒီ rule ကို ၅ ကြောင်းမြောက်မှာ ထည့်ချင်တယ်ဆိုပါစို့။ အသုံးပြုရမယ့် syntax က အောက်ပါအတိုင်းပါ။

iptables -I INPUT 5 -s 192.168.1.11 -d 192.168.1.10 -p tcp --dport 22 -j DROP

-I (အကိုး) **INPUT 5** လို့ သုံးလိုက်တဲ့အတွက် ဒီ rule ဟာ iptables ရဲ့ inbound policy ရဲ့ ၅ ကြောင်းမြောက်မှာ နေရာယူသွားပါလိမ့်မယ်။ ပီးရင် --> service iptables save
ပီးရင် --> service iptables restart ဒါပါပဲ

အခြေခံ concepts တွေ၊ option တွေက ယခင် post တွေမှာ ဖော်ပြပြီးပြီ့ ထည့်ရှုင်းမနေတော့ ဘူးနော်။

+ + [protecting invalid accesses] ++

ကိုယ့် server ဆီ ဝင်လာလေ့ရှိတဲ့ DDOS attack တွေကို linux iptables firewall နဲ့ ဘယ်လို ကာမလဲ ကြည့်ကြဖို့။

အောက်ပါ policy ငှုံးချင်မှု ရေးသွင်းပေးရှုပါပဲ။ **--tcp-flags** ထဲမှာ **SYN,ACK,FIN,RST** စသိဖြင့် ပါပါတယ်။ ဒီနေရာမှာ အကုန်လုံးပါအောင် **ALL** လို့ သုံးထားပါတယ်။ options တွေကို အသေးစိတ်တော့ မရှင်းပြတော့ဘူး။ သိချင် Google ပေါ့ ^_^

၁။ **SYN attack** ကို ကာဖို့

`iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP`

၂။ **Fragment attack** ကို ကာဖို့ `iptables -A INPUT -f -j DROP`

၃။ **XMAS attack** ကို ကာဖို့ `iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP`

၄။ **NULL packets** တွေကို ဖြုတ်ချဖို့

`iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP`

ဒါတွေကို ရေးသွင်းပါးရင်၊ မိမိသုံးမယ့် **ip** တွေ **port** တွေကို ဆက်ဖွင့်ပေးပေါ့။ ဒီလောက်ပါပဲ။

+ + [backup and restore iptables' rules] + +

linux ရဲ့ **firewall** ဖြစ်တဲ့ **iptables** မှာ သတ်မှတ်ထားတဲ့ **rules** တွေကို **backup/restore** လုပ်တဲ့အကြောင်း ပြောကြဖို့။ ဘာ့ကြောင့် **backup and restore** လုပ်ဖို့ လိုအပ်တာလဲ?

၁။ မိမိရဲ့ **config** ကို သိမ်းထားဖို့။

၂။ လွှဲမှားမှု မရှိစေဖို့။

၃။ အချိန်ကုန် သက်သာဖို့။

ဥပမာ ဆိုရရင် --> **server** ၅ လုံးကို **run** ထားပါတယ် ထားပါတော့။ အားမှာ **server** တိုင်းကို **iptables rules** တွေ တူတူ သတ်မှတ်ပေးရမယ်ပေါ့။ ၁ လုံးချင်း ဝင်ပီး **configuration** လုပ်နေရင် ကြာပါတယ်။ နောက် မတူညီမှတွေ မှားယွင်းတာတွေ ရှိနိုင်တယ်။ ဒီတော့ ဘယ်လို စီစဉ်မလဲ?

၁။ **iptables** ကို စိတ်တိုင်းကျတဲ့အခါ **file** တစ်ခုအဖြစ် **backup** လုပ်မယ်။

၂။ အဲဒီ **backup file** ကို တွေား **server** တွေပေါ်က **iptables** ဆီ **restore** လုပ်မယ်။ ဒါဆို **server** အားလုံး **config** တွေလည်းတူ့၊ အချိန်လည်း အကုန် သက်သာပြီပေါ့။

ကဲ ! စလိုက်ရအောင် ...

#backup

ဒီတော့ **iptables** ရဲ့ **rules** တွေကို ဘယ်လို **backup** လုပ်မလဲ?

`iptables-save ipt.bk` ဆိုတဲ့ **command** ကို **run** ပါ။ ဒါဆို current directory အောက်မှာ **ipt.bk** ဆိုတဲ့ file လေး အဖြစ် ရလာပါလိမ့်မယ်။ memory stick ထဲ ကူးရှုပေါ့။

အထဲမှာ ဘာတွေ ရေးထားလဲ၊ သိချင်ရင် `vi ipt.bk` ဆိုတဲ့ command နဲ့ ဖွင့်ကြည့်ပါ။ server ရဲ့ **iptables** ကို သူများ ဘယ်လို configuration ချသွားလဲ သိချင် ဒီနည်း သုံးနိုင်ပါတယ်။ ဒါက **backup** အပိုင်း။

#restore

ဒီတော့ backup file ကို တွေး server မှာ ဘယ်လို restore လုပ်မလဲ?

`iptables-restore ipt.bk` ဆိုတဲ့ command ကို run ပါ။ ဒါဆို iptables ရဲ့ rules ထွေတူဖြစ်သွားပါပြီ။ ပီးရင် --> `service iptables save` ပီးရင် --> `service iptables restart` ဒါပါပဲ

+ + [extra knowledge] + +

လိုတဲ့အခါ သုံးဖို့ ၁၅၅ services, port numbers and protocols -->

ftp >> 20, 21 >> tcp

ssh >> 22 >> tcp

telnet >> 23 >> tcp

smtp >> 25, 465, 587 >> tcp

pop3 >> 110, 995 >> tcp

http >> 80, 443 >> tcp

ping >> 0, 8 >> icmp

dns >> 53 >> tcp and udp

snmp agent >> 161, 162 >> udp ကဲ ! အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(2-8-2018)



LVM - Logical Volume Management

linux မှာ hard disk (HD) ကို ပုံမှန် partition ပိုင်းတာ၊ virtual memory ဖန်တီးတာတွေ ပြီးခဲ့ပါပြီ။ ခုတစ်ခါက LVM နဲ့ logical volume တွေ ဖန်တီးတာကို တင်ပြပေးမှာပါ။ ဘုံးကြောင့် ရှိုးရှိုး partition မပိုင်းပဲ logical volume (LV) တွေ ဖန်တီးမှာလဲ။ ဘုံးကြောင့်လဲ ဆိုတော့ LVM မှာ အား သာချက်တွေ ရှိလိုပါ။

၁။ partition တစ်ခုကို အလွယ်တကူ size တိုးတာ

၂။ partition တစ်ခုကနေ တစ်ခုကို data တွေ မထိခိုက်စေပဲ down time zero နဲ့ ရွေ့ပစ်တာ ဒါတွေ လုပ်လို့ ရပါတယ်။ ကဲ ! စလိုက်ကြစို့။

LV တွေ ဖန်တီးရမှာ အစဉ်လိုက် အသုံးပြုသွားရမယ့် command တွေကတော့ အောက်ပါ အတိုင်းပါ။

၁။ `pvcreate`

၂။ `pvs`

၃။ vgcreate

၄။ vgs

၅။ lvcreate

၆။ lvs

အရင်ဆုံး HD ကို fdisk ဆိုတဲ့ command အသုံးပြုဖြီး partition ပိုင်းရှာမှာ type ID ကို 8e လို့
သတ်မှတ်ပေးလိုက်ပါ။ type ID သတ်မှတ်ပေးတဲ့ အကြောင်းကို ယခင် status မှာ ရေးသား
ခဲ့ဖြီးပါဖြီး ဟုတ်ပြီ။ size 2G စီ ရှိတဲ့ /dev/sdb1 နဲ့ /dev/sdb2 ဆိုတဲ့ partition ၂ ခု ရပြီဆိုပါ
တော့။ ဒီ partition ၂ ခုကို အရင်းပြုဖြီး logical volume (LV) ထွေ ဖန်တီးလိုက်ရအောင်။
၁။ အရင်းဆုံး partition တွေကို physical volume (PV) ထွေ အဖြစ် ပြောင်းလဲရပါမယ်။
partition ၂ ခုကို PV အဖြစ် ပြောင်းလဲထို့ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

pvcreate /dev/sdb1 /dev/sdb2

၂။ ရရှိလာတဲ့ physical volume ၂ ခုကို ပြန်ကြည့်ချင်ရင် pvs လို့ ရှိက် enter ခေါက်ရပါမယ်။ pvs
၃။ စောစောက ရရှိလာတဲ့ physical volume ၂ ခုကို အရင်းပြုဖြီး volume group (VG) ၁ ခု
အောက်ပေးရပါမယ်။ command syntax ကတော့ အောက်ပါအတိုင်းပါ။

vgcreate my_vg /dev/sdb1 /dev/sdb2

my_vg ဆိုတာ **volume group** အမည်ပါ။ မိမိ နှစ်သာက်သလို ပေးနိုင်ပါတယ်။

၄။ ရရှိလာတဲ့ **volume group** ကို ပြန်ကြည့်ချင်ရင် vgs လို့ ရှိက် enter ခေါက်ရပါမယ်။ vgs

၅။ ခုဆိုရင် physical volume >> sdb1 နဲ့ sdb2 နှစ်ခုပေါင်း 4G ရှိတဲ့ my_vg လို့ အမည်ရတဲ့
volume group ၁ ခု ရလာပါပြီ။ ဒီ vg ကနေမှ 1G စီ ရှိတဲ့ logical volume (LV) ၂ ခု ဖန်တီးပါမယ်။
command syntax ကတော့ အောက်ပါအတိုင်းပါ။ lvcreate -n my_lv1 -L 1G my_vg ဒါဆို ၁ ခု
ခု ရပါပြီ။ lvcreate -n my_lv2 -L 1G my_vg ဒါဆို ၂ ခု ရပါပြီ။ my_lv1 နဲ့ my_lv2 က
အမည်တွေပါ။ မိမိ နှစ်သာက်သလို ပေးနိုင်ပါတယ်။

၆။ ရရှိလာတဲ့ logical volume တွေကို ပြန်ကြည့်ချင်ရင် lvs လို့ ရှိက် enter ခေါက်ရပါမယ်။ lvs
logical volume တွေကို ဖန်တီးတဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။

#resize_the_LV

ဒီ logical volume တွေကို size ထပ်တိုး extend လုပ်တဲ့ အပိုင်းဆီ ဆက်သွားကြရအောင်။

ဥပမာ၊ စောစောက my_lv ဟာ 1G နဲ့ မလောက်ဘူး ဆိုပါတော့။ သူကို 1G ထပ်တိုးမယ် ဆိုပါစို့။
lvextend ဆိုတဲ့ command ကို အသုံးပြုရပါတယ်။ အသုံးပြုရမယ့် command syntax ကတော့
အောက်ပါအတိုင်းပါ။ lvextend -L +1G /dev/my_vg/my_lv1

lvs လို့ ရှိက် enter ခေါက်ပြီး ပြန်ကြည့်ပါ။ my_lv1 မှာ 1G ထပ်တိုးနေတာ တွေ့ရပါလိမ့်မယ်။
ကဲ ! ဟုတ်ပြီ။

#formatting

ဒီ logical volume ဖြစ်တဲ့ my_lv1 ကို format ချကြည့်ရအောင်။

ext4 နဲ့ format ချမယ် ဆိုပါစို့။ command syntax ကတေသာ အောက်ပါအတိုင်းပါပဲ။

```
mkfs.ext4 /dev/my_vg/my_lv1
```

xfs နဲ့ format ချမယ် ဆိုပါစို့။ command syntax ကတေသာ အောက်ပါအတိုင်းပါပဲ။

```
mkfs.xfs /dev/my_vg/my_lv1
```

ဒါဆို my_lv1 ကို xfs type နဲ့ format ချပြီးပြီ ဆိုပါစို့။ directory တစ်ခုခုမှာ mount လုပ်ပြီး storage အဖြစ် အသုံးပြုလို့ ရပြီပေါ့။

#mounting

mount လုပ်တဲ့ အပိုင်းလေး ဆက်သွားလိုက်ရအောင်။ current user က root ဆိုပါစို့။ mkdir

storage လို့ရိုက် enter ခေါက်ပြီး current user ရဲ့ home directory အောက်မှာ storage ဆိုတဲ့ အမည်နဲ့ directory တစ်ခု ထောက်လိုက်ပါ။

my_lv1 ကို storage ဆိုတဲ့ directory (folder) ဆို mount လုပ်ဖို့ အသုံးပြုရမယ့် command syntax ကတေသာ အောက်ပါအတိုင်းပါပဲ။ mount /dev/my_vg/my_lv1 /root/storage

ကဲ ! ခုဆို storage ဆိုတဲ့ directory ထဲမှာ data တွေ စတင် သိမ်းနိုင်ပါပြီ။

Logical Volume Management (LVM) အကြောင်းကတေသာ ဒီလောက်ပါပဲ။

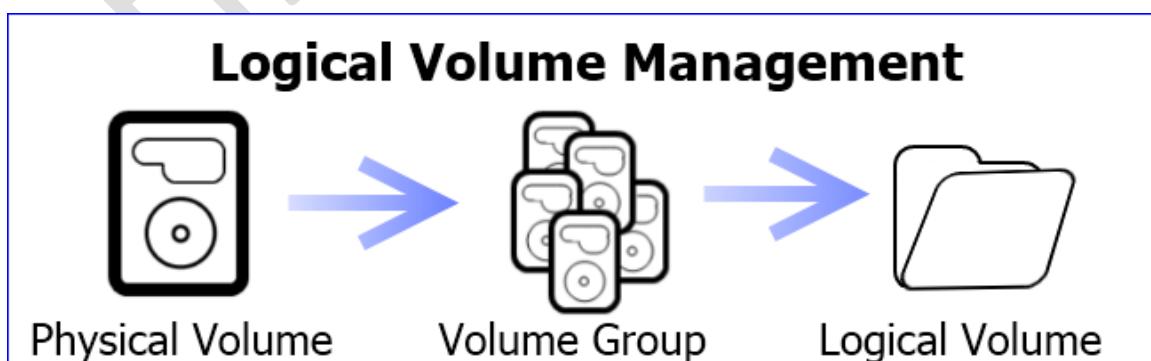
နည်းနည်း ရှုပ်သလို ထင်ရပေါ်ယူ ၃ ခါလောက် ဖတ်ရင်း စမ်းကြည့်မယ်ဆို လွယ်ကူသွားမှာပါ။

pv >> vg >> lv ဒီလို့ တစ်ဆင့်ချင်း သွားတာမို့ လွယ်လွယ်လေးပါနော်။

အားလုံး အဆင်ပြုပါစော့။

Yin Thu

(8.8.2018)



Accessing Network Storage (NFS)

ဒီတစ်ခါ တင်ပြမှာက linux မှာ server ဘက်က share ထားတဲ့ storage တစ်ခုကို မိမိ စက်ရဲ့ သတ်မှတ်ထားတဲ့ directory တစ်ခုဆီ mount လုပ် ချိတ်ဆက်တဲ့အကြောင်းပါ။

server ဘက်က storage ၁ ခုကို share မယ်။ ဒါကို ကိုယ့်စက်ကနေ လုမ်းချိတ်ယူမယ်။ ဒါပါပဲ။

server ဘက်ကနေ mount လုပ်ရမယ့် source လမ်းကြောင်းတွေ option တွေ ပေးထားပါ လိမ့်မယ်။

ဥပမာ

type >> **nfs**

options >> **sync,sec=krb5p**

source >> www.example.com/share/public

keytab >> <http://www.example.com/pub/krb5.keytab> စသဖြင့်ပေါ့။

nfs ဆိုတာက **network file system** ကို ဆိုလိုတာပါ။

sync, sec=krb5p ကတော့ **security option** ပါ။

source ဆိုတာကတော့ ဒီနေရာမှာ **server** ဘက်က ပေးထားတဲ့ **root directory** အောက်က

share အောက်က **public** ဆိုတဲ့ **storage** (www.example.com/share/public) ကို

မိမိဘက်မှာ **mount** လုပ် တပ်ဆင်ပေးရမှာပါ။ **keytab** ဆိုတာကတော့ **security** နဲ့ ဆိုင်တဲ့ **key**

ကို **download** လုပ်ရမယ့် **link** ပါ။

#####

ကဲ ! စလိုက်ကြရအောင်။

၁။ **nfs-utils** ဆိုတဲ့ **package** ကို **install** လုပ်ပါ။ `yum -y install nfs-utils`

၂။ **wget** ဆိုတဲ့ **command line download manager** ကို သွင်းပါ။ `yum -y install wget`

၃။ **keytab** ကို **download** ဆွဲပြီး **root directory** အောက်က **etc** အောက် (`/etc/`) မှာ နေရာ

ချုပါ။ `wget -O /etc/krb5.keytab "http://www.example.com/pub/krb5.keytab"`

၄။ **mkdir** နဲ့ **mount point (directory)** တစ်ခု ဖန်တီးပါ။ `mkdir /root/mountpoint`

၅။ **server** ဘက်က **share** ထားတဲ့ **storage** ကို အောက်ပါ **command** နဲ့ ကြည့်ပါ။

`showmount -e serveraddress` **serveraddress** နေရာမှာ **server** ရဲ့ **hostname** ကို ထည့်ပေးရမှာပါ။ ဥပမာ www.example.com

၆။ **server** ဘက်က **storage** ကို ယာယီ **mount** ဖို့ အတွက် command syntax ကတော့ အောက်ပါအတိုင်းပါပဲ။

`mount -t nfs -o sync,sec=krb5p server:/share/public /root/mountpoint`

reboot လုပ်လိုက်ရင် ပြန်ပျက်ပါတယ်။

ဂူ။ **server** ဘက်က **storage** ကို အမြဲ **permanent mount** ထားချင်ရင် **/etc/fstab** ကို **vi** နဲ့
ဖွင့်ပြီး အောက်ပါအတိုင်း ရေးသွင်းပါ။

`www.example.com:/share/public /root/mountpoint nfs sync,sec=krb5p 0 0`

၈။ ပြီးရင် အောက်ပါ command ၂ ခု ကို run ပေးပါ။

`systemctl enable nfs-secure`

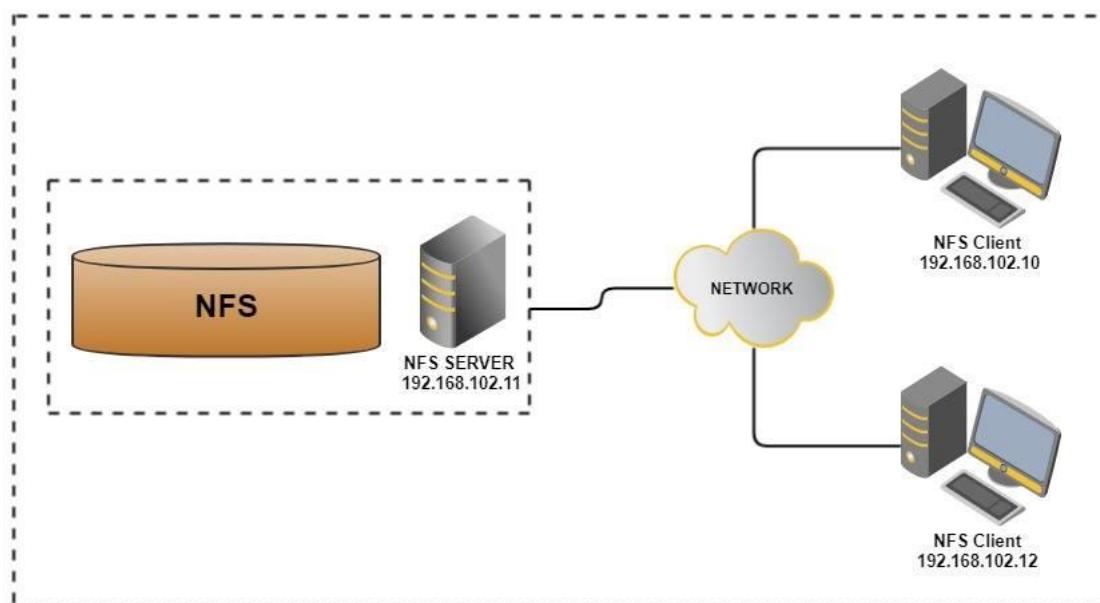
`systemctl restart nfs-secure`

ဒါလိုရင် server ဘက်က ပေးထားတဲ့ storage ကို မိမိဘက်ခြမ်းမှာ တပ်ဆင်ခြင်း ပြီးဆုံးပါပြီ။

အားလုံး အဆင်ပြေပါစေ။

Yin Thu

(10-8-2018)



Accessing SMB (SAMBA) File System

smb ကို linux နဲ့ windows အကြား storage sharing လုပ်ရာမှာ အသုံးများပါတယ်နော်။
ဒီတစ်ခါ ရေးမှာက တွေ့ SMB Server ဘက်ကနေ share ထားတဲ့ storage directory ကို client
ဘက်မှာ ချိတ်ဆက်တဲ့ အကြောင်းပါ။
mounting/mount လုပ်တယ်ဆိုတာ တစ်ဘက်က share ထားတဲ့ storage ကို မိမိ ဘက်ခြေား
client side မှာ တပ်ဆင်တာကို ဆိုလိုတာပါ။

mount လုပ်တာနဲ့ ပတ်သက်လို့ ယခင် post တွေမှာလည်း ရေးခဲ့ဘူးပါပြီ။ mount လုပ်နည်းတွေက
တစ်ခုနဲ့တစ်ခု အချက်အလက်ရေးသွင်းတဲ့ အပိုင်းမှာ အနည်းငယ် လွှဲရှုမှုအပ် အခြေခံ concept
ချင်းက အတူတူပါပဲ။

SMB မှာဆိုရင် mount လုပ်ရာမှာ user name နဲ့ password လေးထည့်ပေးရတာ ၁ ချက် ပိုလာတာ
တစ်ခုပါပဲ။

#####

ကဲ ! စလိုက်ရအောင်။

ဒီလို ချိတ်ဆက်ဖို့အတွက် SMB Server ဘက်ကနေ အောက်ပါ အချက်အလက်တွေကို ပေးထားပါ
လိမ့်မယ်။

၁။ share ထားတဲ့ directory လမ်းကြောင်း

၂။ user name, password နဲ့ domain group

၃။ file system type (cifs) စတာတွေကို ပေးထားပါလိမ့်မယ်။

အဲဒီပေးထားတဲ့ အချက်တွေကို အသုံးပြုပြီး smb server ဘက်က share ထားတာကို မိမိ ဘက်မှာ
mount လုပ်ပေးရမှာပါ။

mount လုပ်ရာမှာ

၁။ ယာယို mount

၂။ permanent mount

၃။ auto mount

ဆိုပြီး ၃ ပိုင်းခွဲ တင်ပြသွားပါမယ်။

မစတင်မီ အရင်ဆုံး cifs-utils ဆိုတဲ့ package ကို အရင်ဆုံး install လုပ်ပါ။ အောက်ပါအတိုင်းပေါ့။

yum -y install cifs-utils

#####

ဥပမာ အနေနဲ့ server ဘက်ကနေ အောက်ပါအတိုင်း share ထားတယ် ဆိုပါစို့။

၁။ /student ကို share ထားတယ်။

၂။ user name က tom, password က redhat, domain group က MYGROUP

၃။ file system type ၂ cifs

ဒီအချက်တွေကို အခြားပြီး mount လုပ်ကြရအောင်။

#####

#ယာယီ_ mount or manual mount

၁။ root အောက်မှာ share ဆိုတဲ့ directory အောက်မှာ student1 အမည့် mount point တစ်ခု ဆောက်ပါ။

mkdir /share

mkdir /share/student1

၂။ user name, password တွေကို file တစ်ခု ဆောက်ပြီး သိမ်းပါ။ touch /etc/secure.smb

၃။ vi /secure.smb လို့ ဖွင့်ပြီး အောက်ပါအတိုင်း ရေးသွင်းပါ။

username=tom

password=redhat

domain=MYGROUP ပြီးရင် **secure.smb** ကို အောက်ပါအတိုင်း permission ပေးပါ။

chmod 600 /secure.smb

၄။ terminal တွင် အောက်ပါအတိုင်း ရေးသွင်းပြီး mount ပါ။

mount -t cifs -o credentials=/etc/secure.smb //server/student /share/student1

ဒါဆို server ဘက်က share ထားတဲ့ student ဆိုတဲ့ directory ကို /share/student1 အောက်မှာ ယာယီ mount နိုင်ပါပြီ။

#####

#permanent_mount or persistent mount

permanent mount ဖို့ အတွက်ကတော့ **/etc/fstab** မှာ အောက်ပါအတိုင်း ဝင်ရေး ပေးရပါမယ်။

//server/student /share/student1 cifs credentials=/ecure.smb 0 0

ဒါဆို permanent mount လုပ်ခြင်း ပြီးဆုံးပါပြီ။

#####

#auto_mount

ဒီးဆုံး **autofs** package ကို install လုပ်ပါ။ yum -y install autofs

၁။ **/etc/auto.master.d** အောက်မှာ **indirect.autofs** လို့ ဆောက်ပေးပါ။

indirect.autofs ထဲမှာ mount point နဲ့ map file လမ်းကြောင်းကို အောက်ပါအတိုင်း ရေးသွင်းပါ။ /share /etc/auto.indirect

ရှင်းရရင် /share >> **mountpoint /etc/auto.indirect >> map file**

J# vi /etc/auto.indirect လို့ ရှိကြပြီး အောက်ပါအတိုင်း ရေးသွင်းပါ။

student1 -fstype=cifs,credentials=/etc/secure.smb ://server/student

၃# systemctl restart autofs လို့ ရှိက် enter ခေါက်ပါ။ systemctl enable autofs လို့ ရှိက် enter ခေါက်ပါ။ ဒါဆို auto mount လုပ်ခြင်း ပြီးဆုံးပါဖြီ။

#####
#####

ကဲ ! SMB server ဘက်က share ထားတဲ့ directory ကို မိမိဘက်ခြမ်းမှာ mount လုပ် ချုတ်ဆက်တဲ့ အကြောင်းကတော့ ဒီလောက်ပါပဲ။

နည်းနည်း ရှုပ်သလို ရှိပေမယ့် ယခင် mount လုပ်နည်း သင်ခန်းစာတွေကို ကြေညာက်ရင် အဆင်ပြုမှာပါနော်။

အားလုံး အဆင်ပြုပါစော်။

Yin Thu

(10-8-2018)



+++ Firewalld (Linux New Generation Firewall) +++

linux system administration နဲ့ပတ်သက်လို့ ဒါက နောက်ဆုံး post ပါ။ ဒါ post လေးနဲ့ နိဂုံးချုပ်ပါရစေ။

red hat, centos နဲ့ fedora linux တို့ရဲ့ firewalld အကြောင်း concept ပါ။ firewall ၁ ခု အနေနဲ့ အကြမ်းအားဖြင့် service တွေကို ဖွင့်/ပိတ် port တွေကို ဖွင့်/ပိတ်၊ စတာတွေကို ဆောင်ရွက်ပေးပါတယ်။ အခါက ၂ inbound/outbound traffic တွေကို ထိန်းချုပ်ဖို့ပေါ့။

အသေးစိတ်ကိုတော့ Iptables (Linux Firewall) မှာ ရေးခဲ့ပြီးပြီမို့ ထပ်မရှင်းတော့ပါဘူးနော်။

firewalld ကို graphical နဲ့ လွယ်ကူစွာ အသုံးပြုနိုင်သလို၊ command line ကနေလည်း အသုံးပြုနိုင်ပါတယ်။

ကဲ ! စလိုက်ရအောင်။

#graphical_user_interface

terminal ထဲမှာ **firewall-config** လိုခိုက် enter ခေါက်လိုက်ပါ။ ဒါဆိုရင် **main firewall configuration screen** ပွင့်လာပါလိမ့်မယ်။

default အနေနဲ့ **zone** မှာ **public zone** ကို သတ်မှတ် ထားပေးပါတယ်။ လက်ရှိ မိမိ **active** ဖြစ်နေတဲ့ **zone** ပေါ့။

services ထဲမှာ မိမိ ဖွင့်ချင်တဲ့ **service** တွေကို အမှန်ခြစ်ပေးရုပါပဲ။ ပိတ်ထားချင်ရင် အမှန်ခြစ်ဖြတ်ပေးရုပါပဲ။

port တွေကို ဖွင့်ပိတ်လုပ်ချင်ရင် **port** ဆိုတဲ့ **tab** ကိုသွားပြီး အမှန်ခြစ်တာ ဖြုတ်တာတွေ ပြုလုပ်ပေးရုပါပဲ။

ပြီးရင် မိမိ ပြောင်းလဲလိုက်တာကို **permanent** သတ်မှတ်ဖို့ အတွက် အပေါ် **menu** မှာ ရှိတဲ့ **Options > Reload Firewalld** ကို နှိပ်ပေးလိုက်ပါ။ လွယ်ကူပါတယ်နော်။

#command_line_interface

ကဲ command line ကနေ အသုံးပြုပုံလေး လေ့လာကြည့်ရအောင်။

၁။ မိမိ **system** မှာ ဘယ် **zone** တွေဟာ **active** ဖြစ်နေလဲ အောက်ပါ **command** နဲ့ ကြည့်ရနိုင်ပါတယ်။ **firewall-cmd --get-active-zones**

၂။ မိမိ ဖွင့်စေချင်တဲ့ service တွေကို အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါတယ်။ ဥပမာ public zone မှာ http, https, ftp စတာတွေကို permanent ဖွင့်ပေးမယ် ဆိုပါစွဲ။

firewall-cmd --add-service={http,https,ftp} --permanent --zone=public

၃။ မိမိ ဖွင့်စေချင်တဲ့ port တွေကို add ပေးမယ်ဆိုပါစွဲ။

firewall-cmd --add-port=port/protocol --zone=public

protocol နေရာမှာ မိမိ အသုံးပြုမယ့် protocol ကို ထည့်ပေးရပါမယ်။

ငါ မိမိ ဖွင့်ထားတဲ့ service တွေကို ပြန်ဖြတ်ချင်ရင် အောက်ပါ command ကို အသုံးပြုရပါမယ်။

ftp service ကို ဖြတ်မယ် ဆိုပါစို့။ `firewall-cmd --remove-service=ftp --zone=public`

၅။ မိမိ ဖွင့်ထားတဲ့ port တွေကို ပြန်ဖြတ်ချင်ရင် အောက်ပါ command ကို အသုံးပြုရပါမယ်။ port

161 ကို ဖြတ်မယ်ဆိုပါစို့။ `firewall-cmd --remove-port=161/protocol --zone=public`

161 ဆိုတာ ဥပမာပေးထားတဲ့ **port number** ပါ။ **protocol** နေရာမှာ **port** နဲ့ သက်ဆိုင်တဲ့ **protocol** ကို ထည့်ပေးရပါမယ်။

၆။ firewall ကို reload လုပ်စေချင်ရင် အောက်ပါ command ကို အသုံးပြုရပါမယ်။

`firewall-cmd --reload`

၇။ firewall ကို restart လုပ်ချင်ရင်တော့ `systemctl restart firewalld` လို့ ရှိက် enter ခေါက်ပေါ့။

`systemctl restart firewalld`

၈။ firewall ကို မသုံးချင်ရင်တော့ အောက်ပါအတိုင်းပေါ့။ `systemctl disable firewalld`

`systemctl stop firewalld`

ကဲ ! Linux ရဲ့ New Generation Firewall ဖြစ်တဲ့ firewalld အကြောင်း concept လေးကတော့ ဒီလောက်ပါပဲနော်။ ပိုပြီး လေ့လာချင်ရင် အောက်ပါ Link လေးမှာ သွားရောက် ဖတ်ရှုပါနော်။

<https://firewalld.org/>

ခုကာစလို့ linux system administration နဲ့ သက်ဆိုင်တဲ့ post တွေ ပြီးဆုံးပါပြီ။

အားလုံး အဆင်ပြေပါစော့။

Yin Thu

(14-8-2018)



BONUS TRACK

+ + [Network Bonding in red hat, centos, fedora linux server] + +

အသုံးတကယ်ဝင်ပြီး Telecom, ISP နဲ့ Enterprise တွေမှာ တကယ် အသုံးပြုနေတာလေး ၁ ခု ပါ။

#intro

ISP တွေ Telecom တွေ bank တွေရဲ့ linux server တွေမှာ network interface တွေကို single interface ၁ ခုတည်းအဖြစ် ပေါင်းစည်းပြီး၊ NIC bonding ကို အသုံးပြုလေ့ ရှိတယ်။ ခုတင်ဆက်မှာ ၁ အဲဒီ NIC Bonding or Network Bonding အကြောင်းပါ။

#purpose

NIC bonding ကို ဘုံးကြောင့် အသုံးပြုသလဲဆိုတော့၊ တစ်ခု fail ဖြစ်သွားရင် နောက် ၁ ခုက hot standby ရှိနေဖို့ အတွက်၊ load balancing အတွက်၊ link integrity ရှိဖို့ အတွက် အသုံးပြုတာပါ။ ဒါမူမဟုတ် speed နဲ့ bandwidth ပိုကောင်းဖို့ အသုံးပြုတာပါ။

#terms

eth0 နဲ့ eth1 ဆိုပြီး network card ၂ card ရှိတယ်ဆိုပါတယ့်။ bond လုပ်မယ်ဆိုရင် eth0 နဲ့ eth1 ကို slave တွေလို့ ခေါပါတယ်။ bond လုပ်ပြီး ရလာတဲ့ interface အသစ်ကိုတော့ bond0 သို့မဟုတ် master လို့ ခေါပါတယ်။

#types

NIC bonding အမျိုးအစား ၆ မျိုး ရှိပါတယ်။

အဲဒီတွေကတော့ - - ->

mode0 >> round-robin policy : packets တွေကို bonded လုပ်ထားတဲ့ NIC တွေကနေ တလုညွစ် transmit လုပ်ပါတယ်။ load balancing နဲ့ fault tolerance ရှိစေပါတယ်။

mode1 >> active backup policy : ဒီ mode မှာ NIC တစ်ခုသာ active ဖြစ်နေပြီး၊ အဲဒီ NIC ပျက်သွားမှ နောက် ၁ ခုက active ဖြစ်လာပါတယ်။ fault tolerance ရှိစေပါတယ်။ အသုံးများပါတယ်။

mode2 >> balance xor policy : load balancing နဲ့ fault tolerance ရှိဖို့အတွက် အသုံးပြုပါတယ်။

mode3 >> broadcast policy : slave interfaces တွေ အားလုံးကနေ transmit လုပ်ပါတယ်။ fault tolerance ရှိစေပါတယ်။

mode4 >> IEEE 802.3ad >> dynamic link aggregation လို့လည်း ခေါပါတယ်။ interface တွေဟာ ပေါင်းစည်းထားတဲ့ group ၁ ခု အနေနဲ့ same speed နဲ့ duplex settings တွေကို

အသုံးပြုပါတယ်။ ဒီ mode ကို သုံးမယ်ဆိုရင် switches တွေ အနေနဲ့ 802.3ad mode ကို support လုပ်ဖို့ enable ဖြစ်ဖို့လိုပါတယ်။
ဒီ mode ကလည်း အသုံးများပါတယ်။

mode5 >> adaptive transmit load balancing : အထူးအထွေ ပြောစရာ မရှိပါဘူး transmit ပိုင်းမှာ load balancing ကို အသားပေးတဲ့ mode ပါ။

mode6 >> adaptive load balancing : အထူးအထွေ ပြောစရာ မရှိပါဘူး receive ပိုင်းမှာ load balancing ကို အသားပေးတဲ့ mode ပါ။

အခဲ့ခဲ့ mode ၆ ခုထဲမှာ အသုံးအများဆုံးက mode1 နဲ့ mode 4 ပါ။

#####

ကဲ ! ဘယ်လို့ bond မယ်ဆိုတဲ့ အကြောင်း ဆက်ကြဖို့။

အောက်ပါအတိုင်း network interface ၂ ခု ရှိတယ် ဆိုပါလို့။

1. eth0

2. eth1

အဲ ၂ card ကို ၁ card ထဲအဖြစ် bond မယ်ပေါ့။

၁။ terminal ထဲမှာ **modprobe --first-time bonding** ဆိုတဲ့ command ကို ရှိက်ပြီး bonding module ကို enable လုပ်ပါ။

၂။ **modinfo bonding** ဆိုတဲ့ command နဲ့ bonding module ရဲ့ information တွေကို ကြည့်နိုင်ပါတယ်။

၃။ **cd /etc/sysconfig/network-scripts/** ဆိုတဲ့ command နဲ့ directory ထဲကို သွားပါ။

၄။ **touch ifcfg-bond0** ဆိုတဲ့ command နဲ့ master interface ကို အောက်ပါ။

၅။ **vi ifcfg-bond0** ကို ဝင်ပြီး အောက်ပါအတိုင်း configuration သွင်းပါ။

```
DEVICE=bond0
NAME=bond0
TYPE=Bond
BONDING_MASTER=yes
IPADDR=192.168.1.150
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
NM_CONTROLLED=no
```

```
BONDING_OPTS="mode=1 miimon=100"
```

ဒီနေရာမှာ BONDING_OPTS="mode=1 ကို mode1 ကို အသုံးပြုမယ်ဆိုတဲ့ သဘောပါ။

```
#####
```

ကဲ ! bond0 ဆိုတဲ့ master card ကိုတော့ configure လုပ်ပြီး၊ slave card ၏ ခုကို ဆက် configure လုပ်ကြပို့။

```
၁။ vi /etc/sysconfig/network-scripts/ifcfg-eth0 ကို ဝင်ပြီး အောက်ပါအတိုင်း အချိုကို ထပ်မံရေးသွင်းပြုနေပါ။
```

```
TYPE="Ethernet"
```

```
BOOTPROTO="none"
```

```
DEFROUTE="yes"
```

```
NAME="eth0"
```

```
ONBOOT="yes"
```

```
MASTER=bond0
```

```
SLAVE=yes
```

```
USERCTL=no
```

```
NM_CONTROLLED=no
```

```
၂။ vi /etc/sysconfig/network-scripts/ifcfg-eth1 ကို ဝင်ပြီး အောက်ပါအတိုင်း အချိုကို ထပ်မံရေးသွင်းပြုနေပါ။
```

```
TYPE="Ethernet"
```

```
BOOTPROTO="none"
```

```
DEFROUTE="yes"
```

```
NAME="eth1"
```

```
ONBOOT="yes"
```

```
MASTER=bond0
```

```
SLAVE=yes
```

```
USERCTL=no
```

```
NM_CONTROLLED=no
```

ခုလို့ configuration သွင်းပြီးတဲ့အခါ `systemctl restart network` ဆိုတဲ့ command နဲ့ network ကို restart လုပ်ပါ။

ပြီးရင် bond လုပ်ထားတဲ့ interface (ifcfg-bond0) အလုပ် လုပ်မလုပ် အောက်ပါ command နဲ့စစ်ဆေးပါ။

cat /proc/net/bonding/bond0

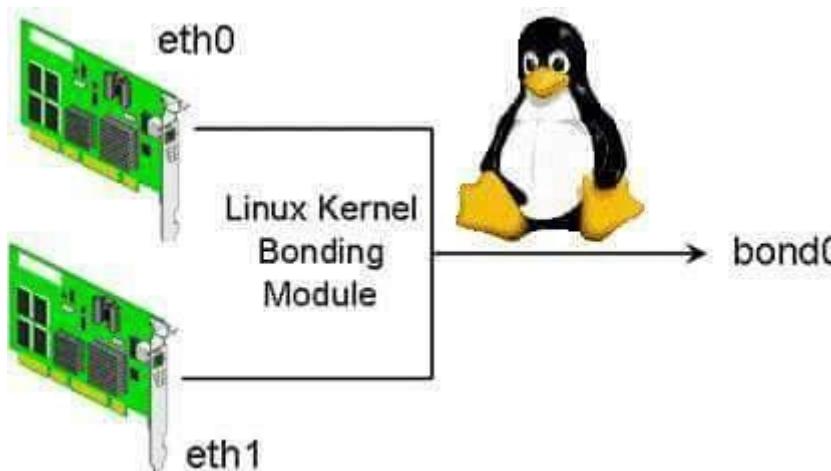
network interfaces တွေရဲ့ list နဲ့ ip တွေကို ကြည့်ဖို့ အောက်ပါ command ကို သုံးပါ။

ip addr

ကဲ ! ခုခို Network Bonding လုပ်ငန်းစဉ် ပြီးဆုံးပါပြီ ...။

အားလုံး အဆင်ပြောပါစေ ^_^

yin thu



++ [UFW : Uncomplicated Fire Wall] ++

တြဲး firewall တွေအကြောင်းလည်း တင်ပေးဘူးပြီမဲ့ firewall နဲ့ ဆိုင်တဲ့ concept တွေကို သိနေကြမယ် ထင်ပါတယ်။ ခုတခါ Linux ရဲ့ Firewall တစ်ခုဖြစ်တဲ့ ufw အောက်ငါး လေ့လာကြည့်ရအောင်။

What is UFW?

UFW ဆိုတာက Debian Linux တွေ အတွက် firewall တစ်ခုပါ။ ubuntu, Kali စတာတွေမှာ အသုံးပြုပါတယ်။ သို့သော်ცြား centos, fedora အပါအဝင်၊ တြဲး linux distro တွေမှာလည်း install လုပ် အသုံးပြုနိုင်တာမို့ ကန်သတ်ချက်တော့ မရှိပါဘူး။

သူကို iptables firewall rules တွေထက် ရေးသားရတာ ပိုမို ရိုးရှင်းတဲ့ ပုံစံနဲ့ ဖန်တီးထားတာပါ။ rules သတ်မှတ်တဲ့ command တွေက human readable ပိုဆန်ပါတယ်။ ဒါ့ကြောင့် သူ့အမည်က uncomplicated firewall တဲ့။

How to install?

apt-get install ufw သို့မဟုတ် **sudo apt-get ufw** လို့ ရိုက် enter ခေါက်ပြီး သွင်းယူနိုင်ပါတယ်။ သို့မဟုတ် **yum install -y epel-release** **yum install -y ufw** ဆိုပြီး သွင်းယူနိုင်ပါတယ်။

How to check ufw?

ufw ဟာ run နေသလား၊ ရပ်နေသလား ဆိုတာကို **ufw status** ဆိုတဲ့ command နဲ့ စစ်ဆေးနိုင်ပါတယ်။

How to run?

ufw ကို run မယ်၊ စက်တက်လာတာနဲ့ run မယ်၊ ဆိုရင် **ufw enable** ဆိုတဲ့ command နဲ့ run နိုင်ပါတယ်။

How to stop?

ufw ကို ပြန်ပိတ်ထားမယ် ဆိုရင် **ufw disable** ဆိုတဲ့ command နဲ့ stop လုပ်နိုင်ပါတယ်။

#Default_rules

ufw ဟာ incoming နဲ့ outgoing traffic တွေကို default အနေနဲ့ အားလုံးကို allow လုပ်ပေးထားပါတယ်။

incoming နဲ့ outgoing ကို default အနေနဲ့ ပိတ်ထားဖို့ command ကတော့ -->

ufw default deny incoming

ufw default deny outgoing

incoming နဲ့ outgoing ကို default အနေနဲ့ ဖွင့်ထားဖို့ command ကတော့ -->

ufw default allow incoming

ufw default allow outgoing

#Connection_and_services

firewall ရဲ့ default rules အရ incoming connection တွေကို block ထားတယ် ဆိုပါစို့။ ssh service ကိုတော့ ဖွင့်ပေးထားချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow ssh

ufw allow 22/tcp

တဖန် ftp service ကို ဖွင့်ပေးထားမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow ftp

ufw allow 21/tcp

တဖန် web service ကို ဖွင့်ပေးထားမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow www

ufw allow 80/tcp

သိမဟတ်

ufw allow proto tcp from any to any port 80,443

တဖန MySQL service ကို ဖွင့်ပေးထားမယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow 3306/tcp

သိမဟတ်

ufw allow from any to any port 3306

တဖန PostgreSQL ကို ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါ အတိုင်းပါ။

ufw allow from any to any port 5432

တဖန mail နဲ့ဆိုင်တဲ့ SMTP ကို ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။

ufw allow 25

တဖန mail နဲ့ဆိုင်တဲ့ POP3 နဲ့ POP3S ကို ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။

ufw allow 110

ufw allow 995

တဖန message service IMAP နဲ့ IMAPS ကို ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။

ufw allow 143

ufw allow 995

#Port_ranges

အထက်မှာ ပါတဲ့ 22, 21, 80 တို့ဆိုတာက သက်ဆိုင်ရာ service ရဲ့ port တွေပါ။ အများစု သိပြီးမယ်ထင်ပါတယ်။

အဲဒီ port တွေကိုမ ဥပမာ 1 ကနေ 1000 အထိ စသည်ဖြင့် range တစ်ခု သတ်မှတ်ပြီး ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။ protocol အမျိုးအစားက tcp ဖြစ်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow 1:1000/tcp

တဖန အဲဒီ port တွေကိုမ ဥပမာ 1000 ကနေ 2000 အထိ စသည်ဖြင့် range တစ်ခု သတ်မှတ်ပြီး ဖွင့်ပေးချင်တယ် ဆိုပါစွဲ။ protocol အမျိုးအစားက udp ဖြစ်တယ် ဆိုပါစွဲ။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow 1:1000/udp

#IP_addresses

မိမိ server ဆီကို သတ်မှတ် ip address ကိုပဲ ချိတ်ဆက်ခွင့်ပြုမယ် ဆိုပါစို့။ ip က 192.168.100.10 ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow from 192.168.100.10

network တစ်ခုလုံးကို ချိတ်ခွင့်ပြုထားချင်တယ်။ 192.168.100.0/24 ဆိုတဲ့ network ကို ခွင့်ပြုမယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw allow from 192.168.100.0/24

#Denying

စောစောက connection တွေ service တွေ port တွေ ip တွေကို allow လုပ် ဖွင့်ပေးတာ။

အဲဒါနဲ့ ပြောင်းပြန် deny လုပ် ပိတ်ထားချင်တာ ဆိုပါစို့။

လွယ်လွယ်လေးပါ **allow** နေရာမှာ **deny** ကို အစားထိုးပြီး command ရိုက်သွားရုံပါပဲ။

ဥပမာ၊ မိမိ server ကနေ သတ်မှတ် ip address ကို ပိတ်ဆိုထားမယ် ဆိုပါစို့။ ip က 192.168.100.10 ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw deny from 192.168.100.10

ဥပမာ၊ မိမိ server မှာက network တွေ အများကြီး ရှိနေတယ်။ eth0, eth1, eth2, eth3, ... စသည်ဖြင့်ပေါ့။

အဲဒီမှာ ip >> 192.168.100.10 ကို ပိတ်ချင်တယ်။ ဘယ်လို ပိတ်ချင်တာလဲဆိုတော့ eth3 ဆိုလာချိတ်မရအောင် ပိတ်ချင်တာ။ အဲဒီအတွက် အသုံးပြုရမယ့် command ကတော့ အောက်ပါအတိုင်းပါ။

ufw deny in on eth3 from 192.168.100.10

#Deleting_rules

ကိုယ်ရေးထားတဲ့ rules တစ်ခုကို ဖျက်ချင်တယ်။ ဥပမာ၊ **ufw allow ssh** ဆိုပြီး ရေးသွင်းခဲ့တာကို ပြန်ဖျက်မယ် ဆိုပါစို့။

ufw delete allow ssh လို့ ရိုက် enter ခေါက်ပြီး ဖျက်ရုံပါပဲ။

သို့မဟုတ်、rules တွေက အရမ်း များနေလို့ နံပါတ်စဉ်နဲ့ ကြည့်ပြီး။ ၁၀ ကြောင်းမြောက် rule ကို ဖျက်ချင်တယ် ဆိုပါစို့။

နံပါတ်စဉ်လိုက် ကြည့်ရန် အောက်ပါ command သုံးပါ။

ufw status numbered

၁၀ ကြောင်းမြောက် rule ကို ဖျက်ရန် အောက်ပါ command သုံးပါ။

ufw delete 10

#Logging

ufw နဲ့ပတ်သက်တဲ့ အချက်တွေကို log လုပ်ပြီး ပြန်ကြည့်နိုင်ပါတယ်။ low, medium, high ဆိုပြီး log level သုံးခဲ့ရှိပြီး default setting ကတေသ့ low ပါ။ အောက်ပါ command ကို သုံးပြီး log တွေကို ဖမ်းယူထားနိုင်ပါတယ်။ **ufw logging**

log တွေကို အောက်ပါအတိုင်း ပြန်ကြည့်နိုင်ပါတယ်။

less /var/log/ufw

#Location_of_rules

သတ်မှတ်ပေးလိုက်တဲ့ rules တွေအနေနဲ့ ဘယ် file တွေထဲကို ဝင်သွားသလဲဆိုတော့၊ **/etc/ufw** ဆိုတဲ့ လမ်းကြောင်းအောက်က **before.rules** နဲ့ **after.rules** ဆိုတဲ့ file တွေထဲကို ဝင်ပါတယ်။

configuration file ကတေသ့၊ **/etc/default/ufw** ဆိုတဲ့ file ပါ။

ဘာတွေ ရေးထားသလဲ သိချင် **vi** နဲ့ဖွင့်ကြည့်ပေါ့။

#Reseting

ကိုယ်ရေးလိုက်တဲ့ rules တွေဟာ ရှုပ်ယူက်ခတ်ပြီး firewall ကို နှစ်မှုလ default အတိုင်း ပြန်ဖြစ်စေချင်တယ် ဆိုပါစို့။ အသုံးပြုရမယ့် command ကတေသ့ အောက်ပါအတိုင်းပါ။

ufw reset

[CONCLUSION] --> firewall တွေကို manage လုပ်မယ်ဆို၊ service တွေ port numbers တွေကို သိမှ အဆင်ပြုပါမယ်။ ဒါ့ကြောင့် အသုံးများတဲ့ service နဲ့ port numbers တွေကို ထည့်ရေးပေးထားတာပါ။ rules တွေ ရေးပုံရေးနည်းကတေသ့ uncomplicated fire wall ဆိုတဲ့ အမည်နဲ့အညီ၊ လွယ်ကူ ရှိုးရှင်းတာမို့ မခက်ခဲပါဘူးနော်။ ထပ်မံ သိရှိလိုသေးရင်တော့ www.google.com ပေါ့နော်။

ဒီလောက်ပါပဲ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu



++ [making, mounting, burning iso file and creating bootable usb from command line] ++

ခုတခါ အလှည့်ကျတာကတော့ iso file တွေနဲ့ဆိုင်တဲ့အကြောင်းပါ။ iso file ဆိုတာ ဘာလဲဆိုတော့ An ISO image file is a snapshot of the data and layout of a CD or DVD, saved in ISO-9660 format. ပါတဲ့။ ဆက်လက်ဖော်ပြုမယ့် အကြောင်းအရာတွေ အတွက် Linux ကို အခြေခံလောက်တော့ သိထားမှ အဆင်ပြေပါမယ်နော်။ စရအောင်။

၁။ Making iso file >> iso file လုပ်ဖို့ အတွက် genisoimage ဆိုတဲ့ tool ကို အောက်ပါအတိုင်း သွေးယူပါ။

Red Hat, Centos, Fedora အတွက်
yum install genisoimage

Ubuntu, Kali အတွက်
apt-get install genisoimage

ဥပမာ /root/Desktop/music ဆိုတဲ့ folder ကို iso image file အဖြစ် ပြောင်းမယ်ဆိုပါစို့။ အသုံးပြုရမယ့် command ၏ အောက်ပါအတိုင်းပါ။
genisoimage -o music.iso /root/Desktop/music

enter ခေါက်လိုက်တဲ့အခါ music.iso ဆိုတဲ့ file လေးရလာပါလိမ့်မယ်။ ဒီနေရာမှာ music.iso သည် မိမိပေးလိုတဲ့ file အမည်ပါ။ နှစ်သက်သလို ပေးနိုင်ပါတယ်။ ဒါပေမယ့် .iso နဲ့ဆုံးဖို့လိုပါလိမ့်မယ်။

၂။ Mounting iso file >> iso file ထဲက data တွေကို ကြည့်ရလိုရင် အဲဒီ iso file ကို folder တခုခုမှာ mount ပေးဖို့ ချိတ်ပေးဖို့လိုပါတယ်။

စောစောကော /root/Desktop/music.iso အမည်ရ file ကို /root/Desktop/testfolder ဆိုတဲ့ နေရာဆီ ချိတ်မယ် ဆိုပါစို့။

mount -o loop -t iso9660 /root/Desktop/music.iso /root/Desktop/testfolder

enter ခေါက်လိုက်တဲ့အခါ **music.iso** သည် **testfolder** ဆီ ချိတ်ပြီးသား ဖြစ်သွားပါလိမ့်မယ်။
cd /root/Desktop/testfolder လို့ ရှိက်ဝင်၊ ls နဲ့ ခေါ် ပြီး **data** တွေကို ကြည့်နိုင်ပါပြီ။

၃။ **Burning iso to CD/DVD >> iso file ကို DVD အဖြစ် Burn နိုင်ပြီး wodim ဆိုတဲ့ **tool** ကို
အောက်ပါအတိုင်း သွင်းယူရပါမယ်။**

Red Hat, Centos, Fedora အတွက်

yum install wodim

Ubuntu, Kali အတွက်

apt-get install wodim

ဥပမာ **music.iso** ကို **DVD** အဖြစ် **Burn** မယ် ဆိုပါစို့။

wodim -eject -tao speed=1 dev=/dev/sr0 -v -musicdisc /root/Desktop/music.iso

Entar ခေါက်လိုက်ရင် iso file ကို DVD အဖြစ် စတင် Burn ပါလိမ့်မယ်။ ဒါ command မှာ
dev=/dev/sr0 သည် **DVD Drive** ကို ရည်ညွှန်စာပါ။ **distro** အလိုက် တခုနဲ့ တခု အမည် တူချင်
မှ တူမှာပါ။ lsblk ဆိုတဲ့ command ကို သုံးပြီး **DVD Drive** အမည်ကို ကြည့်နိုင်ပါတယ်။

၄။ **Creating bootable usb >> linux OS** တခုခဲ့၊ ဥပမာ **fedora OS (Fedora-Workstation-Live-x86_64-29-1.2.iso)** ကို **usb flash drive** ကနေ **boot** ဖြစ်အောင် ဖန်တီးမယ် ဆိုပါစို့။

ဦးစွာ၊ **usb flash drive** ကို **laptop** မှာ တပ်လိုက်ပါ။ အလိုလို **mount** သွားပါလိမ့်မယ်။ ဥပမာ
flash drive name က **myflashdrive** လို့ အမည်ပေးထားတယ်ဆိုပါစို့။
/media/root/myflashdrive ဆိုပြီး **mount** နေတယ် ဆိုပါစို့။ အဲဒီ **flash drive** ကို အောက်
ပါအတိုင်း **umount** ပြန်လုပ်ပေးရပါမယ်။ umount /media/root/myflashdrive
အကယ်၍ ဘာအမည်နဲ့ ဘယ်လို **mount** ထားသလဲ သိလိုရင် **lsblk** ဆိုတဲ့ command နဲ့
ကြည့်နိုင်ပါတယ်။

ကဲ ! **umount** လုပ်ပြီးပြီ ထားပါတွေ့။

flash drive ရဲ့ partition ကို format ချပေးရပါမယ်။ `mkfs.vfat /dev/sdb1 -I`

ဒီနေရာမှာ /dev/sdb1 သည် **flash drive** ရဲ့ **partition** ပါ။ တခုနဲ့တခု မတူ ကဲ ပြားနေနိုင်တယို့
fdisk -l ဆိုတဲ့ command နဲ့ စစ်ဆေးပြီးမှ **format** ချပါ။

ဒါနို **flash drive** သည် **bootable usb** လုပ်ဖို့ အသင့်ဖြစ်ပါပြီ။ **dd** ဆိုတဲ့ command ကို အသုံးပြုပြီး အောက်ပါအတိုင်း **bootable usb** ပြု လုပ်ပါ။

```
dd if=/root/Desktop/Fedora-Workstation-Live-x8_64-29-1.2.iso of=/dev/sdb1 bs=4M  
&& sync
```

ကဲ ! command line ပေါ်မှာ iso file နဲ့ သက်ဆိုင်တဲ့ အကြောင်းတွေကတော့ ဒီလောက်နဲ့ နားပါရစေ။

အားလုံး အဆင်ပြုပါစေ။

PS --> [Post ကို နှစ်သက်ရင် Like, Comment, Share လုပ်နိုင်ပါတယ်။ Page က တင်သမျှကို မြင်ရဖို့ Page ကို Like အပြင် see first လုပ်ပေးထားဖို့ လိုအပ်ပါတယ်]

Yin Thu

(14-2-2019)



ဘယ် process က cpu နဲ့ memory ကို အများဆုံး ယူသုံးနေသလဲ? ပြီးတော့ စက်ပေါ့သွားအောင်၊ Memory Cache ကို ဘယ်လို ရှင်းလင်းမလဲ?

ခုတင်ပြုမယ့် topic လေးက ရှိုးရှင်းပေမယ့် အရမ်း အသုံးဝင်ပါတယ်။ ဥပမာ NOC တွေမှာ Monitoring ကြည့်နေရင်း Memory usage က တဖြည့်ဖြည့် တက်လာတယ်။ အရမ်းတက်လာတဲ့အခါ Monitoring Screen မှာ warning alarm တွေပြုမယ်၊ ဒီအခါ အခါ တင်ပြုမယ့် နည်းလမ်းက အသုံးဝင်လာပါတယ်။

၁။ ပထမ အနေနဲ့ ဘယ် process တွေက cpu ကို အများဆုံး ယူသုံးနေသလဲ စစ်ဆေးပါမယ်။ command ကတော့ အောက်ပါအတိုင်းပါ။

```
ps -aux --sort=-%cpu | awk 'NR<=10{print $1 " " $2 " " $3 " " $11}'
```

ဒါဆို့ cpu ကို အများဆုံး သုံးစွဲနေတဲ့ process ကိုးခုကို ပြုသပေးပါလိမ့်မယ်။ cpu ကို သုံးတာ များလို့ warning တက်တာကတော့ ရှားပါတယ်။

J။ ဒုတိယ အနေနဲ့ ဘယ် process တွေက memory ကို အများဆုံး ယူသုံးနေသလဲ စစ်ဆေးပါမယ်။

```
ps -aux --sort=-%mem | awk 'NR<=10{print $1 " " $2 " " $4 " " $11}'
```

ဒါဆို့ memory ကို အများဆုံး သုံးစွဲနေတဲ့ process ကိုးခုကို ပြုသပေးပါလိမ့်မယ်။ memory ကို သုံးတာ များလို့ warning တက်တာကတော့ အဖြစ်များပါတယ်။

+ + [clearing Memory Cache] + +

Memory နေရာလွှတ်တွေ available space တွေ တက်လာဖို့ ram ကို clear လုပ်ဖို့လိုအပ်ပါတယ်။ ဒါမှုလည်း server ရဲ့ work load လျော့မှာပါ။ အဲလို လုပ်ရာမှာ level 3 ခု ရှိပြီး production မှာဆိုရင် level 1 ကိုပဲ အသုံးပြုသင့်ပါတယ်။ level 2 နဲ့ 3 က ကိုယ့် personal စက်မှာပဲ အသုံးပြုသင့်ပါတယ်။ memory ကို ရှင်းလင်းဖို့ အတွက် command ကတော့ အောက်ပါအတိုင်းပါ။

```
sync; echo 1 > /proc/sys/vm/drop_caches
```

မိမိ ကိုယ်ပိုင် စက်ဆိုရင် echo နောက်မှာ 3 လို့ ထည့်သွင်းနိုင်ပါတယ်။

တစ်ခု ပြောစရာရှိတာက နေ့စဉ် Memory က Alarm တက်နေတာဆိုရင် နေ့စဉ် ပုံမှန် ရှင်းပေးဖို့လိုလာပါပြီ။ ဥပမာ နေ့စဉ် မနက် ၆ နာရီတိုင်း Memory ကို ရှင်းမယ် ဆိုပါစို့။ command ကို crontab ထဲ သွားပေးလိုက်ရင် အဆင်ပြေပါတယ်။

၁။ crontab -e လို့ရှိက်

၂။ i ကို စွဲပြီး insert mode ထဲဝင်

၃။ * * * * sync; echo 1 > /proc/sys/vm/drop_caches လို့ဖြည့်

၄။ Esc :wq နဲ့ save လုပ်ပြီး ပြန်ထွက်

ဒါဆို နေ့စဉ် မနက် ၆ နာရီတိုင် Memory ကို ပုံမှန် ရှင်းပေးသွားပါလိမ့်မယ်။

ကဲ ! ခုတာခါတော့ ဒီလောက်ပါပဲ။

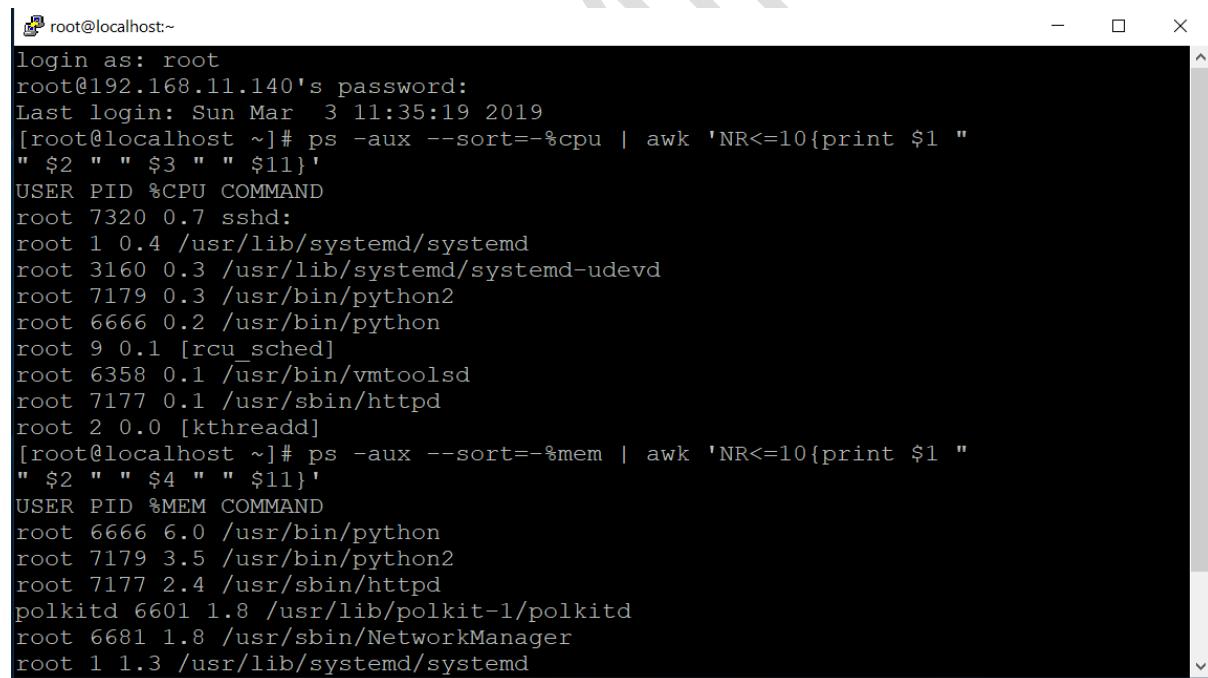
အရေးအကြောင်းကြံတဲ့ အခါ ဒီနည်းလေးကို သတိရကြပေါ့။

အားလုံး အဆင်ပြေပါစေ။

PS --> [Post ကို နှစ်သက်ရင် Like, Comment, Share လုပ်နိုင်ပါတယ်။ Page ကနေ တင်သမျက်မြင်ရဖို့ Page ကို Like အပြင် see first လုပ်ပေးထားဖို့ လိုအပ်ပါတယ်]

Yin Thu

(13-2-2019)



```
root@localhost:~# login as: root
root@192.168.11.140's password:
Last login: Sun Mar  3 11:35:19 2019
[root@localhost ~]# ps -aux --sort=-%cpu | awk 'NR<=10{print $1 " " $2 " " $3 " " $11}'
USER PID %CPU COMMAND
root 7320 0.7 sshd:
root 1 0.4 /usr/lib/systemd/systemd
root 3160 0.3 /usr/lib/systemd/systemd-udevd
root 7179 0.3 /usr/bin/python2
root 6666 0.2 /usr/bin/python
root 9 0.1 [rcu_sched]
root 6358 0.1 /usr/bin/vmtoolsd
root 7177 0.1 /usr/sbin/httpd
root 2 0.0 [kthreadd]
[root@localhost ~]# ps -aux --sort=-%mem | awk 'NR<=10{print $1 " " $2 " " $4 " " $11}'
USER PID %MEM COMMAND
root 6666 6.0 /usr/bin/python
root 7179 3.5 /usr/bin/python2
root 7177 2.4 /usr/sbin/httpd
polkitd 6601 1.8 /usr/lib/polkit-1/polkitd
root 6681 1.8 /usr/sbin/NetworkManager
root 1 1.3 /usr/lib/systemd/systemd
```

For Linux System Administrator Interview

The Linux Boot Process

- 1.BIOS
- 2.Boot Loaders
- 3.Kernel/OS
- 4.Runlevels

1.BIOS

- .Basic Input/Output System
- .Special firmware
- .Primary purpose is to find and execute the boot loader
- .Performs the POST (Power-On Self Test)
- .Knows about bootable devices (Hard drive,USB drive,DVD drive..etc.)

2.Boot loaders

- .LILO (Linux Loader)
- .GRUB (Grand Unified Bootloader,Replaced LILO)
- .Boot loaders start the operating system

Initial RAM Disk

- .initrd (initial RAM disk)
- .Temporary file system that is loaded from disk and stored in memory.
- .Contains helpers and modules required to load the permanent OS file system.

/boot

- .Contains the files required to boot Linux

.initrd
.kernel (vmlinuz)
.boot loader configuration

Kernel Ring Buffer

- . Contains messages from the Linux kernel
- . dmesg (command can see that messages)
- . /var/log/dmesg

System Runlevels

- . 0 Shutdown(or halt) the system
- . 1 Single-user mode (Used for maintenance)
- . 2 Multi-user mode without networking
- . 3 Multi-user with networking
- . 4 Udefined
- . 5 Multiuser mode with networking and GUI
- . 6 Reboot the system

Init

- . /etc/inittab (Before, assign Runlevel)
- . Now Used by systemd

Systemd

- . Uses targets instead of runlevels.

```
cd /lib/systemd/system  
ls -l runlevel5.target  
systemctl get-default  
systemctl set-default graphical.target
```

Changing runlevels or targets

.telinit Runlevel

telinit 5

.systemctl isolate Target

systemctl isolate graphical.target

Rebooting

telinit 6

systemctl isolate reboot.target

reboot

shutdown [options] time [message]

shutdown -r 15:30 "rebooting!"

shutdown -r +5 "rebooting soon!"

shutdown -r now

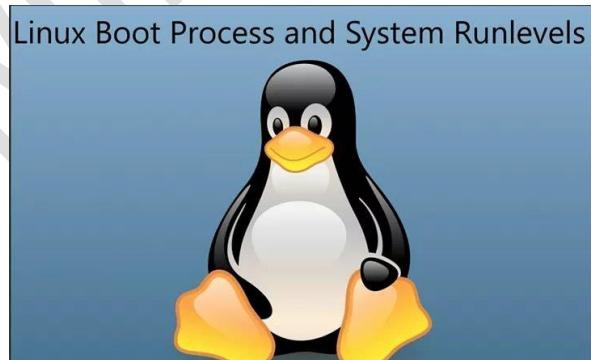
Poweroff

telinit 0

systemctl isolate poweroff.target

poweroff

credit - HappyLinux (facebook page)



red hat linux ကို subscription နဲ့ သံဃားကြမယ်

မသိသေးတဲ့သူတွေ အတွက်ပါ။ <https://www.redhat.com> မှာ account ဖွင့်ပြီးရင် အဲအကောင့်နဲ့ Red Hat ကို subscription လုပ်ပြီး ၁ နှစ် သုံးခွင့် ရပါတယ်နော်။ ၁ နှစ်ပြည့်သွားရင်လည်း renew လုပ်ပြီး ဆက်သုံးနိုင်ပါတယ်။

၁။ account ဘယ်လို ဖွင့်ရမလဲဆိုတော့ page ရဲ့ အပေါ်ညာဘက်က log in ဆိုတာလေးကို နှိပ်လိုက်ရင်၊ login နဲ့ register ဆိုပြီး ကျေလာပါမယ်။ အဲအထဲက register ကနေ account ဖွင့်ပါ။

၂။ subscription ဘယ်လို ဖြည့်ရမလဲ ဆိုတော့ >> install လုပ်ပြီးလို့ boot ပြန်တက်လာရင် subscription manager ဆိုတာ တွေ့လိမ့်မယ်။ next ကို နိုင်၊ အဲကနေ ကိုယ့်ရဲ့ redhat account name (အမည်ဖြစ်သည် mail မဟုတ်) နဲ့ password ကို ဖြည့်ပေးရုံပါ။

ဒါမှာမဟုတ်၊ GUI က Application >> system tools >> subscription manager ကနေလည်း လုပ်နိုင်ပါသေးတယ်။

၃။ red hat ကို download လုပ်ဖို့ Link -->

<https://developers.redhat.com/products/rhel/download/>

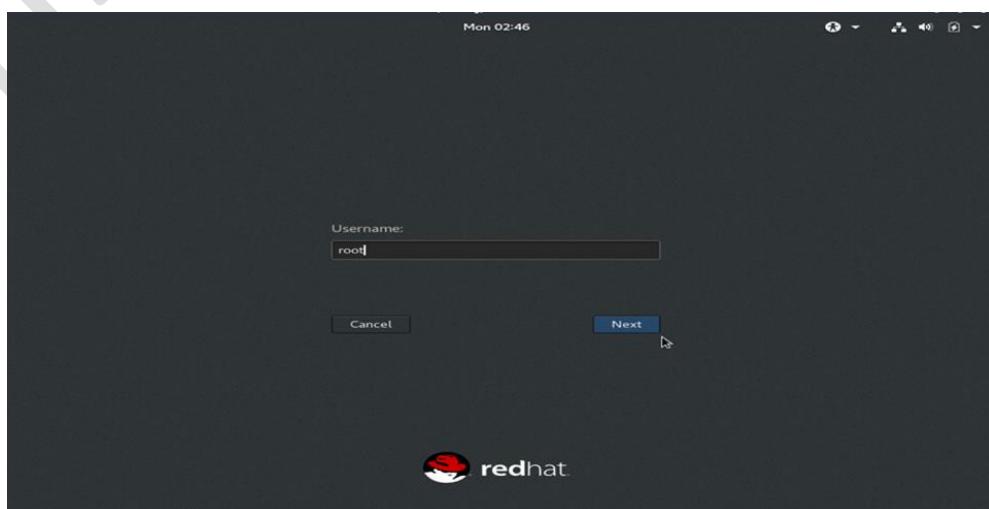
၄။ download မလုပ်ချင်ရင် နီးစပ်ရပ် Software အရောင်းဆိုင်ကနေဝါယ်ပါ။

၅။ red hat ဆီကနေ၊ ပိုက်ဆံနဲ့ ဝယ်တာမဟုတ်လို့ red hat ရဲ့ support တော့ မရဘူးပေါ့နော်။ တခုခုဆို community ကိုပဲ အားကိုးရမယ်ပေါ့။

ဒါမယ့် သုံးပျော်ပါတယ်နော်။ အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(10-12-2018)



[**nmcli (connect wifi via terminal)**] --> ခုတခါ CLI : Command Line Interface ကန် wifi ချိတ်ကြည့်ရအောင်။

၁။ **nmcli radio wifi on** : ဒါက စက်ရဲ့ wireless network card ကို enable လုပ်တာပါ။

၂။ **nmcli dev wifi list** : ပတ်ဝန်းကျင်က hot spot list ကို ကြည့်တာပါ။

ကဲ ! VALIANT SNS ဆိုတဲ့ hot spot ကို ချိတ်ရအောင်

၃။ **nmcli dev wifi connect VALIANT** လို့ ရှိက် **Tab key** ကို နှိပ်ပါ။ ဒါဆို **nmcli dev wifi connect VALIANT\ SNS** လို့ auto ဖြည့်သွားပါလိမ့်မယ်။

၄။ **nmcli dev wifi connect VALIANT\ SNS password 12345678** : ပြီးရင် enter ခေါက်ပါ။

၅။ Device 'wlan0' successfully activated with စသဖြင့်ပေါ်လာရင် အောင်မြင်စွာ ချိတ်ဆက်ပြီးပြီပေါ့

၆။ ပြန်ဖြတ်ဖို့ကတော့ **nmcli radio wifi off** လို့ ရှိက်ရုံပါ

မှတ်ချက်။ ။ **nmcli** မရှိလှုင် online မှ သွင်းယူပါ။ **apt-get -y install networkmanager** သို့မဟုတ် **yum -y install networkmanager** ။

အားလုံး အဆင်ပြုပါစေ။

Yin Thu

(27-11-2018)