Linux Permissions

Linux System ပေါ်က File တွေ Directory တွေဟာ permission ကိုယ်စီနဲ့အလုပ်လုပ်ကြပါတယ်။ ဒီ Permission တွေနဲ့ သူတို့ကို ပိုင်ဆိုင်ဖို့ သတ်မှတ်ပေးထားတဲ့ Owner User, Owner Groups တွေဟာ သတ်မှတ်ထားတဲ့ permission တွေအတိုင်း Access ရယူရပါတယ်။

Permission (၃) မျိုး ရှိပါတယ်။

- Read
- Write
- Execute တို့ပဲဖြစ်ပါတယ်။

File တစ်ဖိုင်မှာ ရှိတဲ့ စာတွေ ၊ အထဲမှာ ရေးသားထားတဲ့ program တွေကို ဖတ်ကြည့်ချင်တဲ့ အချိန်မှာ ဖတ်ကြည့်မယ့် User အတွက် Read Permission လိုအပ်ပါတယ်။

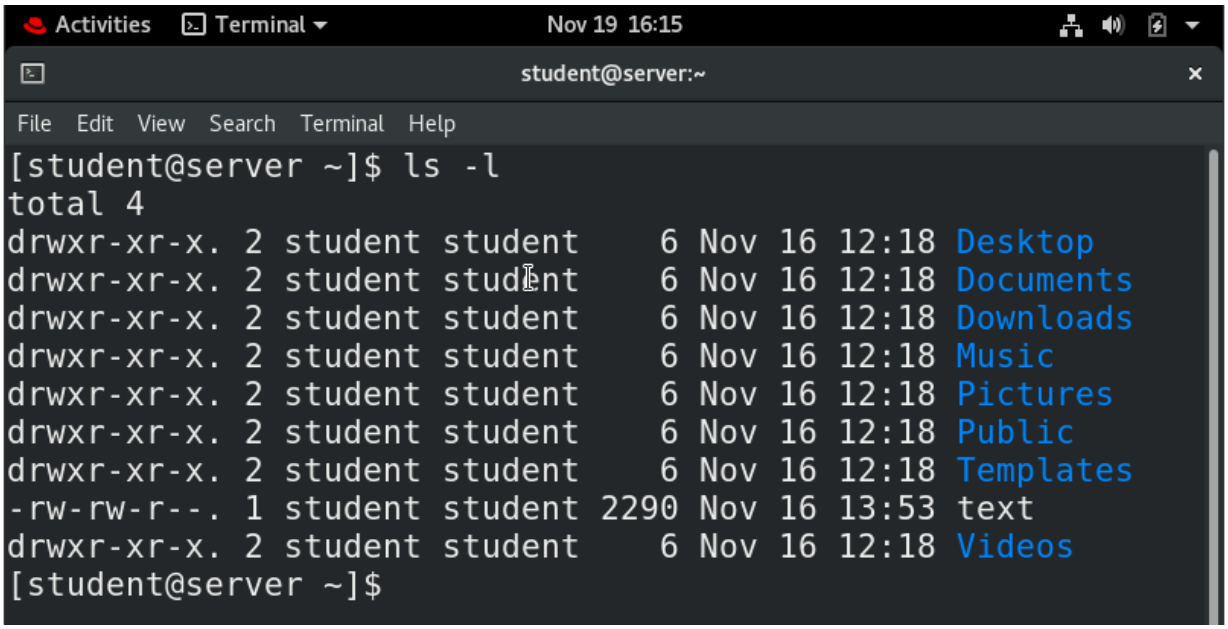
File တစ်ဖိုင်မှာရှိတဲ့ စာတွေ program တွေကို ပြန်လည်ရေးသားလိုတဲ့အခါ ဒီ file ကို ပြန်လည်ပြုပြင် modified လုပ်မယ့် user အတွက် Write Permission လိုအပ်ပါတယ်။

File တစ်ဖိုင်ဟာ ရိုးရိုး ရေးသားထားတဲ့ file ပဲဖြစ်မယ်ဆိုရင် ကျွန်တော်တို့အနေနဲ့ execute permission ကို ပေးလို့ရပါတယ်။ ဒါပေမယ့် လိုအပ်ပါဘူး။ Execute ဟာ Program ရေးသားထားတဲ့ file ကို ပေးရမယ့် permission မျိုးဖြစ်ပါတယ်။ Execute လုပ်ချင်တဲ့ User ဟာ ဒီ program file ကို execute access ရှိမှ run နိုင်မှာဖြစ်ပါတယ်။

Permission တွေကို အတိုကောက်အနေနဲ့

Permission	shortform	Decimal form
Read	r	4
Write	w	2
Execute	x	1

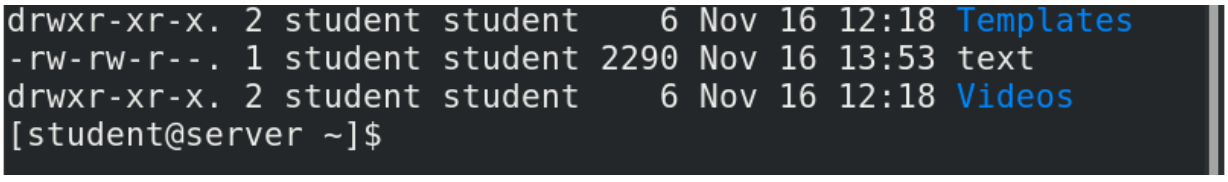
ဒီလိုသတ်မှတ်နိုင်ပါတယ်။ လက်ရှိ Data တွေရဲ့ Permission တွေကို ကြည့်ချင်ရင် ls -l နဲ့ကြည့်နိုင်ပါတယ်။



အထက်ပါပုံ အရ

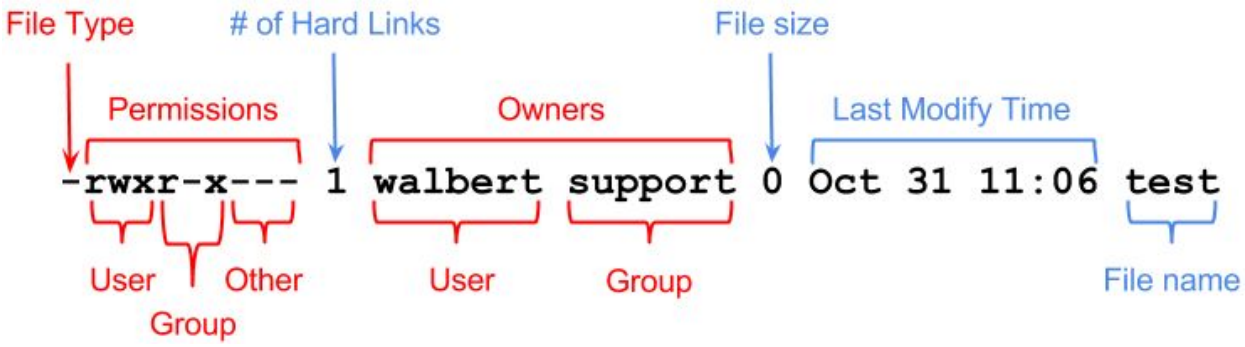
drwxrwxr-x. 2 student student 6 Nov 16 12:18 Desktop

ဒီ ကောင်သည် Directory ဖြစ်ကြောင်း ဖော်ပြထားပါတယ်။ တကယ်လို file ဆိုရင် ထိပ်ဆုံး d ဆိုတဲ့ နေရာမှာ - လေးနဲ့ပြမှာဖြစ်ပါတယ်။



အထက်ပါပုံမှာဆိုလျှင် text ဆိုတာသည် file ဖြစ်ကြောင်းကို - နဲ့ပြထားတာဖြစ်ပါတယ်။

-rw-rw-r--. 1 student student 2290 Nov 16 13:53 text



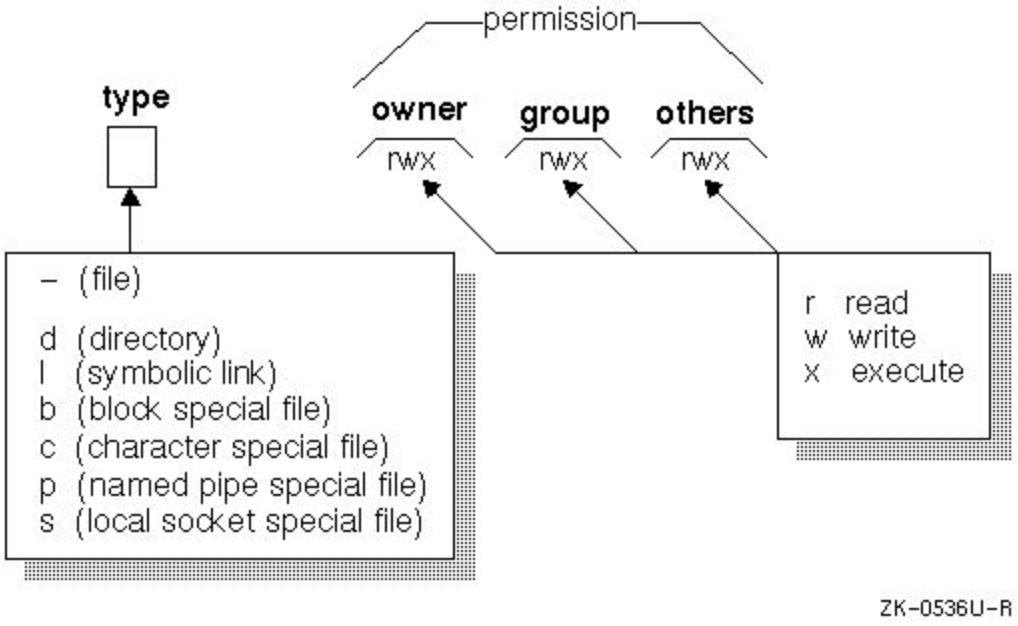
အထက်ပါ diagram အရ ဆိုရင် rwxr-x--- ဆိုပီး permission ပေးထားတာကိုတွေ့ရပါလိမ့်မယ်။

ပထမ rwx က walbert user အတွက်ဖြစ်ပြီး၊ ဒုတိယ r-x ဆိုတာ group အတွက်ဖြစ်ပါတယ်။ ဘယ် group လဲဆိုတော့ support အတွက်ဖြစ်ပါတယ်။ r-x မှာ w ကျန်နေတာကိုတွေ့ရပါလိမ့်မယ်။ w ဖြုတ်ထားလို့ (-) ဖြစ်နေတာပါ။ ဒါဆိုရင် ဒီ support group ကတော့ read နဲ့ execute ပဲ ရပါလိမ့်မယ်။ write ခွင့်မရှိပါဘူး လို့ဆိုလိုတာဖြစ်ပါတယ်။

နောက်တစ်ခု other နေရာမှာ --- သုံးခု ပေးထားပါတယ်။ other ဆိုတာက owner user လည်းမဟုတ်ဘူး။ group ထဲမှာလည်းမပါဝင်တဲ့ user ကို ပြောတာဖြစ်ပါတယ်။ --- သုံးခုဆိုတော့ read, write, execute သုံးခုစလုံးမပေးဘူးလို့ဆိုလိုတာဖြစ်ပါတယ်။ ဒါဆိုရင် other user အတွက် ဘာမှ access မရနိုင်ပါဘူး။

NOTE

နောက်တစ်ခုက Directory ကို permission သတ်မှတ်တဲ့အခါ execute အမြဲတမ်းပေးရပါတယ်။ ဒါမှ သူ့ရဲ့ directory ထဲကို ဖွင့်ပြီး အထဲက data တွေကို access လုပ်နိုင်မှာဖြစ်ပါတယ်။



ဒါဆိုရင် Decimal တွေနဲ့ permission သတ်မှတ်ပုံကိုလေ့လာကြမယ်။

	4	2	1	
0	-	-	-	no permissions
1	-	-	x	only execute
2	-	w	-	only write
3	-	w	x	write and execute
4	r	-	-	only read
5	r	-	x	read and execute
6	r	w	-	read and write
7	r	w	x	read, write and execute

အထက်ပါပုံအရဆိုရင် read = 4 , write = 2 , execute = 1 လို့ပြောခဲ့ပါတယ်။ ဒါဆိုဒီသုံးခုကို ပေါင်းပြီးပေးရမှာဖြစ်ပါတယ်။

ဥပမာ- ကျွန်တော်တို့ test ဆိုတဲ့ file လေးကို permission ပေးကြည့်ရအောင်။ user owner ကို read, write, execute သုံးခု အပြည့်ပေးမယ်။ ဒါဆိုရင် 4+2+1 = 7

7 owner test

နောက်တစ်ခု group ကို read, နဲ့ write ပဲပေးကြမယ်။ 4+2 = 6

76 owner,group test

နောက်တစ်ခု others ကို read ပဲပေးရအောင်။ read = 4

764 owner,group,other test4

ဒါဆိုရင် အဖြေက

```
-rwxrw-r--. 1 student student    0 Nov 19 17:01 test
```

ပထမ owner ကိုပေးတော့ permission အပြည့် 7 ပေးလိုက်ပါတယ်။ ဒါဆိုရင် 7 = 4+2+1 ဖြစ်ပါတယ်။

ဒုတိယ group ကိုပေးတော့ permission က read, write ပဲပေးပါတယ်။ ဒါဆိုရင် 6 =4+2 ဖြစ်ပါတယ်။

တတိယ other ကိုပေးတော့ permission က read ပဲပေးပါတယ်။ ဒါဆိုရင် read က 4 ဖြစ်ပါတယ်။

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

အိုကေ ဒါတွေက ရိုးရိုး permission တွေပဲ ရှိပါသေးတယ်။ Linux ပေါ်မှာ special permission တွေရှိပါသေးတယ်။

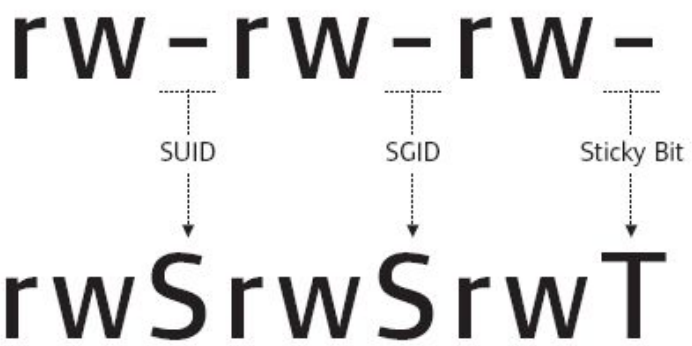
Linux ပေါ်မှာ Special Permission (၃) မျိုးရှိပါတယ်။

SetUID = file ပေါ်မှာသတ်မှတ်ပြီးအလုပ်လုပ်ရတဲ့ permission အမျိုးအစားဖြစ်ပါတယ်။ execute လုပ်ခွင့်ရှိတဲ့ file တစ်ဖိုင်ကို setuid ဆိုတဲ့ special permission ပေးထားမယ်ဆိုရင်

အဲ့ဒီ file ကို လာ run တဲ့ဘယ် user မဆို file ရဲ့ owner အနေနဲ့ပဲ run ခွင့်ရှိတာဖြစ်ပါတယ်။ အတိုကောက်အနေနဲ့ s လို့သတ်မှတ်ပါတယ်။

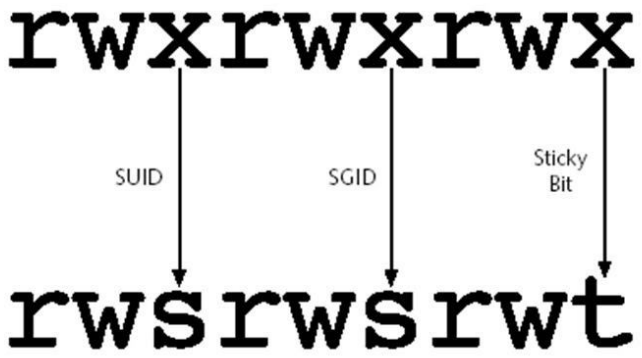
SetGID = file (or) Directory ကိုပေးရတဲ့ special permission အမျိုးအစားဖြစ်ပါတယ်။ setgid ပေးလိုက်မယ်ဆိုရင် ဒီ directory ထဲမှာ ဘယ် user မဆို နောက်ထပ် ထပ်ဆောက်မယ့် သူတို့ ရဲ့ file တွေ directory တွေ အကုန်လုံးသည် အခုလက်ရှိ directory မှာပေးထားတဲ့ GroupOwner အတိုင်းပဲ တည်ဆောက်စေမှာဖြစ်ပါတယ်။ အတိုကောက်အနေနဲ့ s လို့သတ်မှတ်ပါတယ်။

Sticky = ဒါလည်း directory ကိုပေးရတာဖြစ်ပါတယ်။ အပေါ်က setGID ပေးလိုက်တဲ့ dir ထဲမှာ ရှိတဲ့ file တွေ dir တွေသည် တည်ဆောက်လိုက်တဲ့ user အပေါ်မူတည်ပီး owner မတူဘဲ group တွေတူနေမှာဖြစ်ပါတယ်။ ဒီ လို group တွေတူနေတယ်ဆိုရင် user အချင်းချင်းက သူတို့နဲ့မသက်ဆိုင်တဲ့ တခြား user ရဲ့ data တွေဖျက်ခွင့်ရှိနေမှာ ဖြစ်ပါတယ်။ ဒိလို ဖျက်ဆီးခွင့်မရှိအောင် (သို့မဟုတ်) owner ကသူပိုင်ဆိုင်တဲ့ file ကိုပဲဖျက်နိုင်အောင် sticky ဆိုတဲ့ special permission ပေထားခြင်းဖြင့် ကာကွယ်နိုင်ပါတယ်။ အတိုကောက်အနေနဲ့ t လို့သတ်မှတ်ပါတယ်။



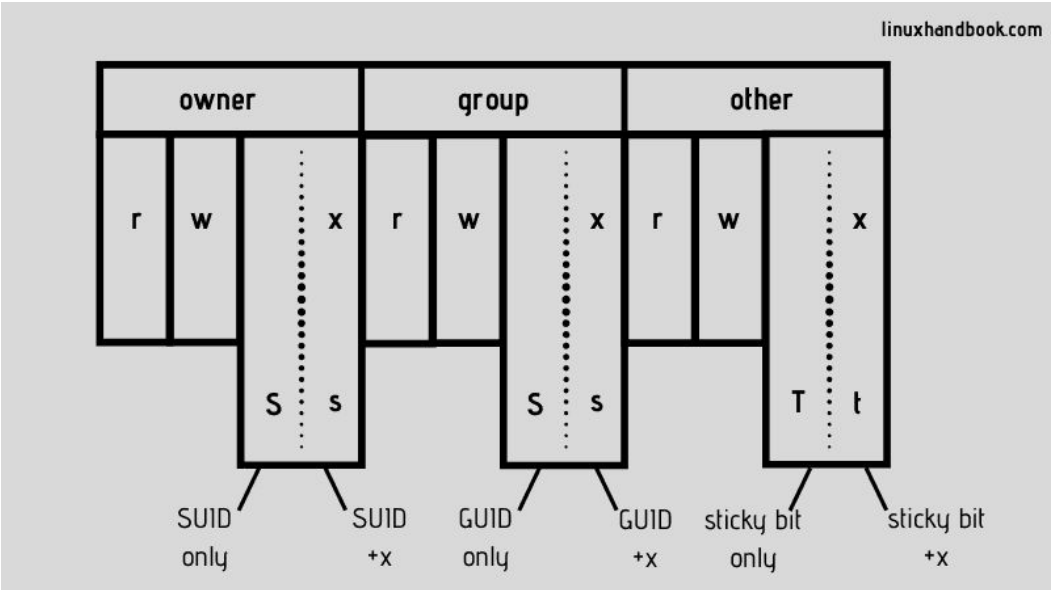
အထက်ပါ diagram လေးကို လေ့လာကြည့်မယ်ဆိုရင် Special Permission တွေက ထည့်စရာနေရာမရှိပါဘူး။ ဒါကြောင့် execute နေရာမှာလိုက်ပြီးထည့်သွားပေးတာဖြစ်ပါတယ်။ execute ကို ဖျောက်လိုက်တာမဟုတ်ပါဘူး execute ကို ဖုံးလိုက်တာပဲဖြစ်ပါတယ်။ အထက်ပါပုံမှာဆိုရင် SetUID ဆိုရင် owner permission နေရာမှာထည့်ပါတယ်။ setGid permission ဆိုရင် group permission မှာထည့်ပီး sticky ဆိုရင်တော့ other user permission နေရာမှာထည့်ထားတာကို တွေ့ရပါမယ်။

အိုကေ S အကြီး ဘာလို့ဖြစ်နေသလဲဆိုတော့ အရင် normal permission မှာ execute (x) မရှိတဲ့ အတွက် ဖြစ်ပါတယ်။



အထက်ပါပုံမှာ ကြည့်မယ်ဆိုရင် special permission ကို သုံးခုစလုံးပေးလိုက်ပါတယ်။ အရင် permission မှာ တုန်းက execute (x) ရှိခဲ့လို့ (s) အသေးနဲ့ special permission ဝင်သွားတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

အောက်ကပုံကို ထပ်လေ့လာကြည့်ရင် ပိုပြီးရှင်းသွားမှာဖြစ်ပါတယ်။



အထက်ပါပုံမှာ ဆိုရင် x မရှိရင် S အကြီးနဲ့ ဝင်မှာ ဖြစ်ကြောင်းနဲ့ other မှာဆိုရင် x မရှိလို့ T အကြီးနဲ့ ဝင်မှာဖြစ်ကြောင်းပြထားတာတွေ့ရပါမယ်။

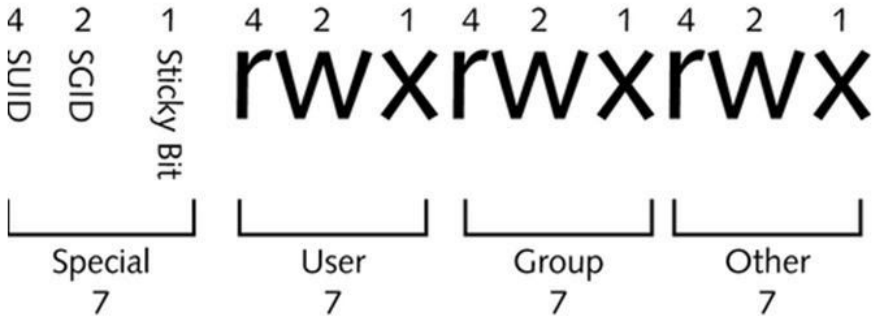


Figure 5-9: Numeric representation of regular and special permissions

အထက်ပါ ပုံမှာကြည့်မယ်ဆိုရင် Special Permission ကို decimal နဲ့ လည်း သတ်မှတ်နိုင်ပါတယ်။

SetUID = 4

SetGID = 2

Sticky = 1 ဖြစ်ပါတယ်။

ဥပမာ - Directory တစ်ခုကို Special Permission ဖြစ်တဲ့ setgid, sticky ပေးချင်တယ်ဆိုပါစို့ ဒါဆိုရင် အရင်က ရိုးရိုး permission ပါပြောင်းလဲချင်တယ်ဆိုပါစို့

- **Special Permission** **setgid=2, sticky=1===3**
- **owner ကို read, write,execute** **read=4, write=2, execute=1 === 7**
- **Group ကို read, execute** **read=4, execute=1 === 5**
- **Other ကို read, execute ပေးမယ်ဆိုပါစို့** **read=4, execute=1 === 5**
- **နောက်ဆုံးအဆင့်အနေနဲ့ အားလုံးကို ယူပြီး 3755 လို့သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။ ဒါဆိုရင် directory ရဲ့ permission ကဘယ်လိုဖြစ်သွားမလဲ**

Before = drwxr-xr-x directory

= 0 7 5 5 directory

After adding special = drwxr-sr-t directory

permission = 3 7 7 5 directory

အနီရောင် highlight လေးနဲ့ ဖော်ပြပေးထားတာကတော့ special permission ပဲဖြစ်ပါတယ်။

```
drwxrwsr-t  2 cops cops     4096 Nov 19 17:51  test/
```

အထက်ပါပုံက တော့ special permission (setgid, sticky) ပေးထားတဲ့ ပုံစံလေးပဲဖြစ်ပါတယ်။

Defaults Permission

Special Permission အပြင် Linux ပေါ်မှာ Default Permission တွေရှိပါတယ်။ umask ဆိုတာ တကယ် actual permission ဖြစ်လာအောင်လုပ်ပေးမှာဖြစ်ပါတယ်။

```
cops@security:~$ umask
0002
cops@security:~$
```

```
root@security:~# umask
0022
```

အထက်ပါ ပုံနှစ်ပုံကို ကြည့်မယ်ဆိုရင် normal user က umask တန်ဖိုးရယ် root user အတွက် umask တန်ဖိုးနှစ်ခုရှိပါတယ်။ ဒါဆိုရင် သူတို့ကဘယ်လိုအလုပ်လုပ်သလဲ အပေါ်က normal user နဲ့အရင်ကြည့်ရအောင်။ cops user ရဲ့ umask တန်ဖိုးက 0002 လို့ပြောထားပါတယ်။ အရှေ့ က 0 တစ်လုံးက special permission အတွက်ဖြစ်ပြီး 002 ကတော့ owner, group, other စတဲ့ permission တွေဘဲဖြစ်ပါတယ်။

Directory base permission	7	7	7	File base permission	6	6	6
Umask	-0	0	2	Umask	-0	0	2
Actual permission	7	7	5	Actual permission	6	6	4
	(rwxrwxr-x)				(rw-rw-r--)		

အထက်ပါပုံမှာဆိုရင် directory အတွက် permission က full permission က 777 ဖြစ်ပြီး file အတွက်ကတော့ 666 ဖြစ်ပါတယ်။ ဒါကို umask တန်ဖိုးနဲ့ နုတ်လိုက်တော့ထွက်လာတဲ့အဖြေတွေက 775 နဲ့ file အတွက်ဆိုရင် 664 ဖြစ်ကြောင်းတွေ့ရပါတယ်။ ထွက်လာတဲ့အဖြေက Actual Permission ပဲဖြစ်ပါတယ်။

ကျွန်တော်တို့ file တစ်ဖိုင်ဆောက်မယ်။ Directory တစ်ခုဆောက်မယ်ဆိုရင် ဒီ actual permission အတိုင်းပဲ ဆောက်ပေးတာဖြစ်ပါတာဖြစ်ပါတယ်။ ဒါကို defaults permission လို့သတ်မှတ်ပါတယ်။