[root@servera ~]# **mkdir /class**
[root@servera ~]# ls -ld /class/

d**rwx r-x r-x**. 2 root root 6 Jul 17 18:43 /class/
7 5 5

[root@servera ~]# **touch apple.txt**
[root@servera ~]# ls -l apple.txt

-**rw- r-- r--**. 1 root root 0 Jul 17 18:43 apple.txt
6 4 4

UMASK

UMASK in Linux or Unix systems is known as User Mask or it is also called as User file creation Mask. This is a base permission or default permission when a new file or folder is created in the Linux machine. It gets involved in each and every step when a new file or directory gets created.

the default settings are not changed, files are created with the access mode 666 and directories with 777. In this example:

The default umask for the root user is 022 result into default directory permissions are 755 and default file permissions are 644.

The default umask 002 used for local user. With this mask default directory permissions are 775 and default file permissions are 664.

For directories, the base permissions are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).

CMD:	**umask**	
SYNTAX:	**# umask**	→Display umask value in numeric notation
	# umask -S	→ Display umask value in symbolic notation
	# umask <umask value>	→ Set Umask Value temporary

ROOT USER:

[root@servera ~]# umask
0 0 2 2
Umask Value
Special Permission Bit

[root@servera ~]# umask -S
u=rwx,g=rx,o=rx

[root@servera ~]# **mkdir kali1 ; touch test1**
[root@servera ~]# **ls -ld kali1**

d**rwx r-x r-x**. 2 root root 6 Jul 17 19:00 kali1
7 5 5

[root@servera ~]# **ls -l test1**

-**rw- r-- r--**. 1 root root 0 Jul 17 19:00 test1
6 4 4

- **how to calculate directory permission and file permission**

  Standard base directory permission (777) – umask value
  Standard base file permission (666) – umask value

Directory Permission			
-	7	7	7
	0	2	2
	7	5	5

Directory Permission			
-	rwx	rwx	rwx
	---	-w-	-w-
	rwx	r-x	r-x

File Permission			
-	6	6	6
	0	2	2
	6	4	4

File Permission			
-	rw-	rw-	rw-
	---	-w-	-w-
	rw-	r--	r--

LOCAL USER

[root@servera ~]# **su - sushmita**

[sushmita@servera ~]$ umask

0022

[sushmita@servera ~]$ **bash**

[sushmita@servera ~]$ **umask**

0 0 0 2

Umask Value
Special Permission Bit

Directory Permission			
-	7	7	7
	0	0	2
	7	7	5

Directory Permission			
-	rwx	rwx	rwx
	---	---	-w-
	rwx	rwx	r-x

File Permission			
-	6	6	6
	0	0	2
	6	6	4

File Permission			
-	rw-	rw-	rw-
	---	---	-w-
	rw-	rw-	r--

[sushmita@servera ~]$ **mkdir data1; touch apple1**

[sushmita@servera ~]$ **ls -ld data1**

d**rwx rwx r-x**. 2 sushmita sushmita 6 Jul 17 19:04 data1
7 5 5

[sushmita@servera ~]$ **ls -l apple1**

-**rw- rw- r--**. 1 sushmita sushmita 0 Jul 17 19:04 apple1
6 4 4

[sushmita@servera ~]$ exit

exit

[sushmita@servera ~]$ exit

logout

CHANGE UMASK VALUE

METHOD: TEMPERORY

SYNTAX:

umask <umask value>

EG:

[root@servera ~]# **umask 444**

[root@servera ~]# umask

0444

Directory Permission			
-	7	7	7
	4	4	4
	3	3	3

Directory Permission			
-	rwx	rwx	rwx
	r--	r--	r--
	-wx	-wx	-wx

File Permission			
-	6	6	6
	4	4	4
	2	2	2

File Permission			
-	rw-	rw-	rw-
	r--	r--	r--
	-w-	-w-	-w-

[root@servera ~]# **mkdir kali2 ; touch test2**

[root@servera ~]# **ls -ld kali2**
d**-wx –wx -wx**. 2 root root 6 Jul 17 19:11 kali2
3 3 3

[root@servera ~]# **ls -l test2**
-**-w- -w- -w-**. 1 root root 0 Jul 17 19:11 test2
2 2 2

[root@servera ~]# **umask 033**
[root@servera ~]# **umask**

0033

Directory Permission			
-	7	7	7
	0	3	3
	7	4	4

Directory Permission			
-	rwx	rwx	rwx
	---	-wx	-wx
	rwx	r--	r--

File Permission			
-	6	6	6
	0	3	3
	6	3	3

File Permission			
-	rw-	rw-	rw-
	---	-wx	-wx
	rw-	-wx	-wx

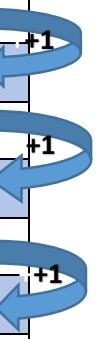
[root@servera ~]# **mkdir kali3 ; touch test3**
[root@servera ~]# ls -ld kali3
d**rwx r-- r--**. 2 root root 6 Jul 17 19:14 kali3
7 4 4

[root@servera ~]# ls -l test3

-**rw- r- -r--**. 1 root root 0 Jul 17 19:14 test3
6 4 4

As the result of file is '**633'** but the permission is set to '**644**' bcoz on the odd bit execute permission comes, and file never gets execute permission by default.
So, in result bit **'1'** is added to the odd bit as odd bit has execute permission

Bit	Permission
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwx



	File Permission		
	6	3	3
+	0	1	1
	6	4	4

	File Permission		
	rw-	-wx	-wx
+	---	--x	--x
	rw-	r--	r-

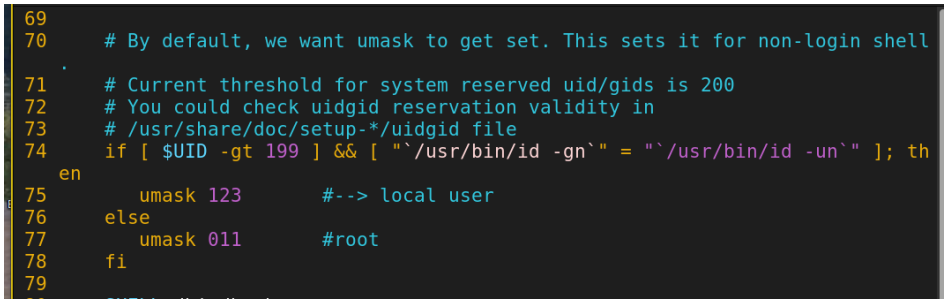
[root@servera ~]# umask

0033

METHOD PERMANENT

Edit global login program file /etc/bashrc to change all users umask value

[root@servera ~]# **vim /etc/bashrc**



```
69
70      # By default, we want umask to get set. This sets it for non-login shell
71      # Current threshold for system reserved uid/gids is 200
72      # You could check uidgid reservation validity in
73      # /usr/share/doc/setup-*/uidgid file
74      if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; th
        en
75          umask 123       #--> local user
76      else
77          umask 011       #root
78      fi
79
```

[root@servera ~]# **cat -n /etc/bashrc | head -79 | tail -6**

```
74      if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
75        umask 123       # → local user
76      else
77        umask 011       # →root
78      fi
79
```

[root@servera ~]# umask

0033

[root@servera ~]# **bash**
[root@servera ~]# **umask**

0011

LOCAL USER

[root@servera ~]# **su - sushmita**

[sushmita@servera ~]$ umask

0022

[sushmita@servera ~]$ bash
[sushmita@servera ~]$ **umask**

0123 → *updated from /etc/bashrc file*

[sushmita@servera ~]$ exit

[sushmita@servera ~]$

exit

[sushmita@servera ~]$

logout

CREATING NEW USER

[root@servera ~]# useradd tom

[root@servera ~]# su - tom

[tom@servera ~]$ umask

0022

[tom@servera ~]$ bash

[tom@servera ~]$ **umask**

0123 → *updated from /etc/bashrc file*

exit

[tom@servera ~]$

logout

CHANGE UMASK FOR SPECIFIED USER SINGLE USER

→ **edit in users login program file, present in users home directory .bashrc**

[root@servera ~]# tail -1 /etc/passwd

tom:x:5005:5007::/home/tom:/bin/bash

[root@servera ~]# ls -a /home/tom

. .. .bash_history . bash_logout .bash_profile .bashrc .cache .mozilla

[root@servera ~]# **vim /home/tom/.bashrc**

[root@servera ~]# tail -1 /home/tom/.bashrc

umask 111

[root@servera ~]# su - tom

[tom@servera ~]$ **umask**

0111

Directory Permission			
-	7	7	7
	1	1	1
	6	6	6

Directory Permission			
-	rwx	rwx	rwx
	-x	--x	--x
	rw-	rw-	rw-

File Permission			
-	6	6	6
	1	1	1
	5	5	5

File Permission			
-	rw-	rw-	rw-
	-x	--x	--x
	r-x	r-x	r-x

As the result bit is odd in files permission and in odd bit execute permission comes so '1' is added to odd bit to get the default permission

File Permission			
+	5	5	5
	1	1	1
	6	6	6

File Permission			
-	r-x	r-x	r-x
	-x	--x	--x
	rw-	rw-	rw-

[tom@servera ~]$ **mkdir red; touch xyz**

[tom@servera ~]$ **ll**

d**rw-rw-rw-**. 2 tom tom 6 Jul 17 19:37 red

-**rw- rw- rw-**. 1 tom tom 0 Jul 17 19:37 xyz
6 6 6

[tom@servera ~]$ exit

logout

Set the default permission for user tom as
File → r-- r-- r--
Dir → r-x r-x r-x

→ Calculate the umask value for the given default permission

Directory Permission			
-	7	7	7
	2	2	2
	5	5	5

Directory Permission			
-	rwx	rwx	rwx
	-wx	-wx	-wx
	r-x	r-x	r-x

File Permission			
-	6	6	6
	2	2	2
	4	4	4

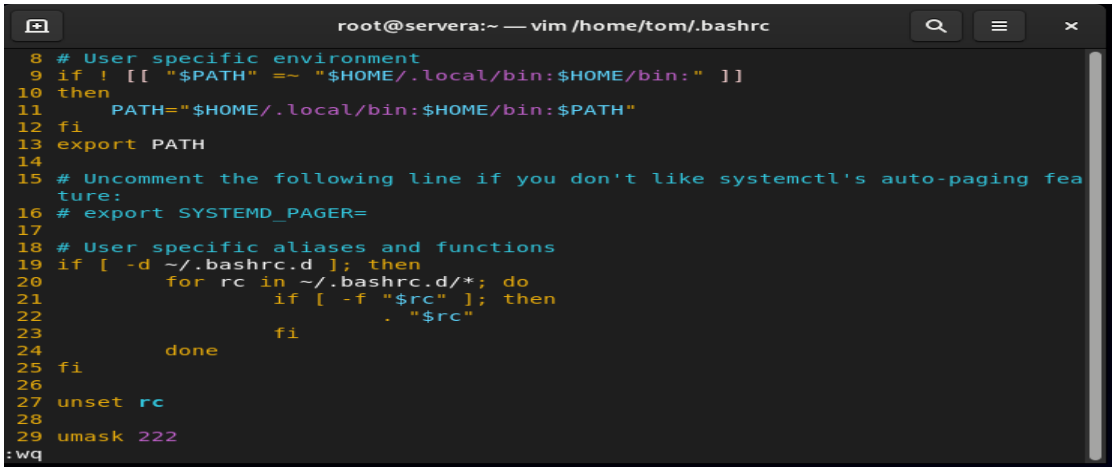
File Permission			
-	rw-	rw-	rw-
	-wx	-wx	-wx
	r--	r--	r--

[root@servera ~]# **ls -a /home/tom**

. .bash_history .bash_profile .cache red

.. .bash_logout .bashrc .mozilla xyz

[root@servera ~]# **vim /home/tom/.bashrc**



[root@servera ~]# **tail -1 /home/tom/.bashrc**

umask 222

[root@servera ~]# **su - tom**

[tom@servera ~]$ **umask**

0 222

[tom@servera ~]$ **mkdir red2 ; touch xyz2**

[tom@servera ~]$ **ls -ld red2 xyz2**

d**r-xr-xr-x**. 2 tom tom 6 Jul 17 19:44 red2
5 5 5

-**r--r--r--.** 1 tom tom 0 Jul 17 19:44 xyz2
4 4 4

[tom@servera ~]$ exit

logout

Set the default permission for user tom as
File → r-- --- ---

Dir → r-x --- ---

Directory Permission			
-	7	7	7
	2	7	7
	5	0	0

Directory Permission			
-	rwx	rwx	rwx
	-wx	rwx	rwx
	r-x	---	---

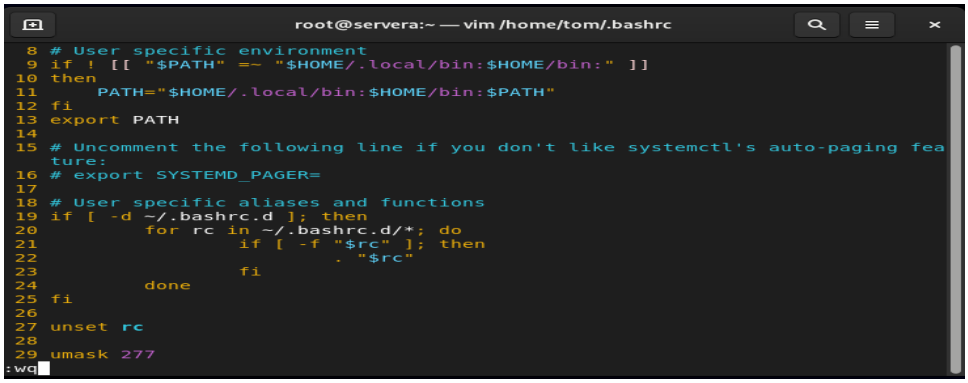
File Permission			
-	6	6	6
	2	6	6
	4	0	0

File Permission			
-	rw-	rw-	rw-
	-wx	rw-	rw-
	r--	---	---

As the directory's and files umask gets differ so in such case use the umask value of directory for default permission

umask → Directory → 277

[root@servera ~]# **vim /home/tom/.bashrc**



[root@servera ~]# **tail -1 /home/tom/.bashrc**

umask 277

[root@servera ~]# **su - tom**

[tom@servera ~]$ **umask**

0277

[tom@servera ~]$ **mkdir red3; touch xyz3**

[tom@servera ~]$ **ls -ld red3 xyz3**

d**r-x --- ---**. 2 tom tom 6 Jul 17 19:49 red3
  5 0 0

-**r-- --- ---**. 1 tom tom 0 Jul 17 19:49 xyz3
  4 0 0

[tom@servera ~]$

logout

SYMBOLIC METHOD

Set the default permission for user tom as

File → rw- --- ---
Dir → rwx --- ---

Directory Permission			
-	7	7	7
	0	7	7
	7	0	0

Directory Permission			
-	rwx	rwx	rwx
	---	rwx	rwx
	rwx	---	---

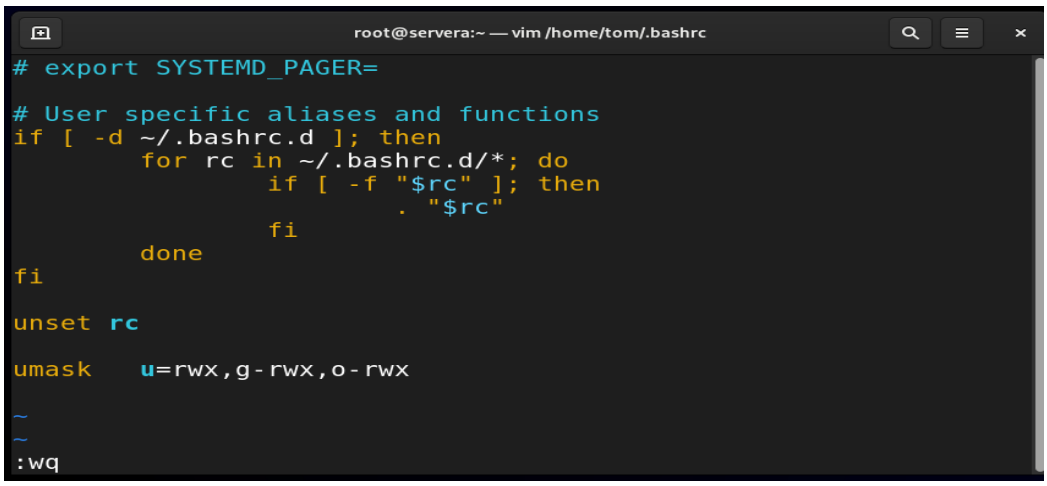
File Permission			
-	6	6	6
	0	6	6
	6	0	0

File Permission			
-	rw-	rw-	rw-
	---	rw-	rw-
	rw-	---	---

As the directory's and files umask gets differ so in such case use the umask value of directory for default permission
For Symbolic method of umask use the result permission of directory

U=rwx g= null o=null

[root@servera ~]# vim /home/tom/.bashrc



[root@servera ~]# **tail -1 /home/tom/.bashrc**

umask u=rwx,g-rwx,o-rwx

[root@servera ~]# su - tom

[tom@servera ~]$ bash

[tom@servera ~]$ **umask**

0077

[tom@servera ~]$ **umask -S**

u=rwx ,g= ,o=

[tom@servera ~]$ **mkdir red4; touch xyz4**

[tom@servera ~]$ **ls -ld red4 xyz4**

d**rwx --- ---**. 2 tom tom 6 Jul 17 19:56 red4
7 0 0

-**rw-** **--- ---**. 1 tom tom 0 Jul 17 19:56 xyz4
6 0 0

exit

[tom@servera ~]$

logout