Introduction aux systèmes d'exploitation

D.TOUAZI FAYCAL

MAITRE DE CONFÉRENCES

UNIVERSITÉ M'HAMED BOUGARA – BOUMERDES

f.touazi@univ-boumerdes.dz

Les droits d’accès

f.touazi@univ-boumerdes.dz

Linux attribue à chaque fichier/répertoire trois droits d'accès:

- **Des droit d'accès pour le propritaire (User)**
- **Des droit d'accès pour un groupe d'utilisateurs (Group)**
- **Des droit d'accès pour les autres utilisateurs dans le système (Other)**

Les droits possibles

- r = Read (permission de lecture)
- w = Write (permission d'écriture)
- x = Execute (permission d'exécution)
- - = Absence de la permission

Signification des droits sur un fichier

Les fichiers sont utilisées pour stocker des données donc:

- **r = Read : On peut lire le contenu du fichier**
- **w = Write : On peut modifier le contenu du fichier**
- **x = Execute : On peut exécuter le contenu de fichier; des instructions pour un fichier binaire ou des commandes pour un script**

Attention:

- **Droit de suppression du fichier fait partie aux droits de son répertoire**

Signification des droits sur un répertoire

Les répertoires sont utilisées pour stocker des fichiers et des répertoires donc:

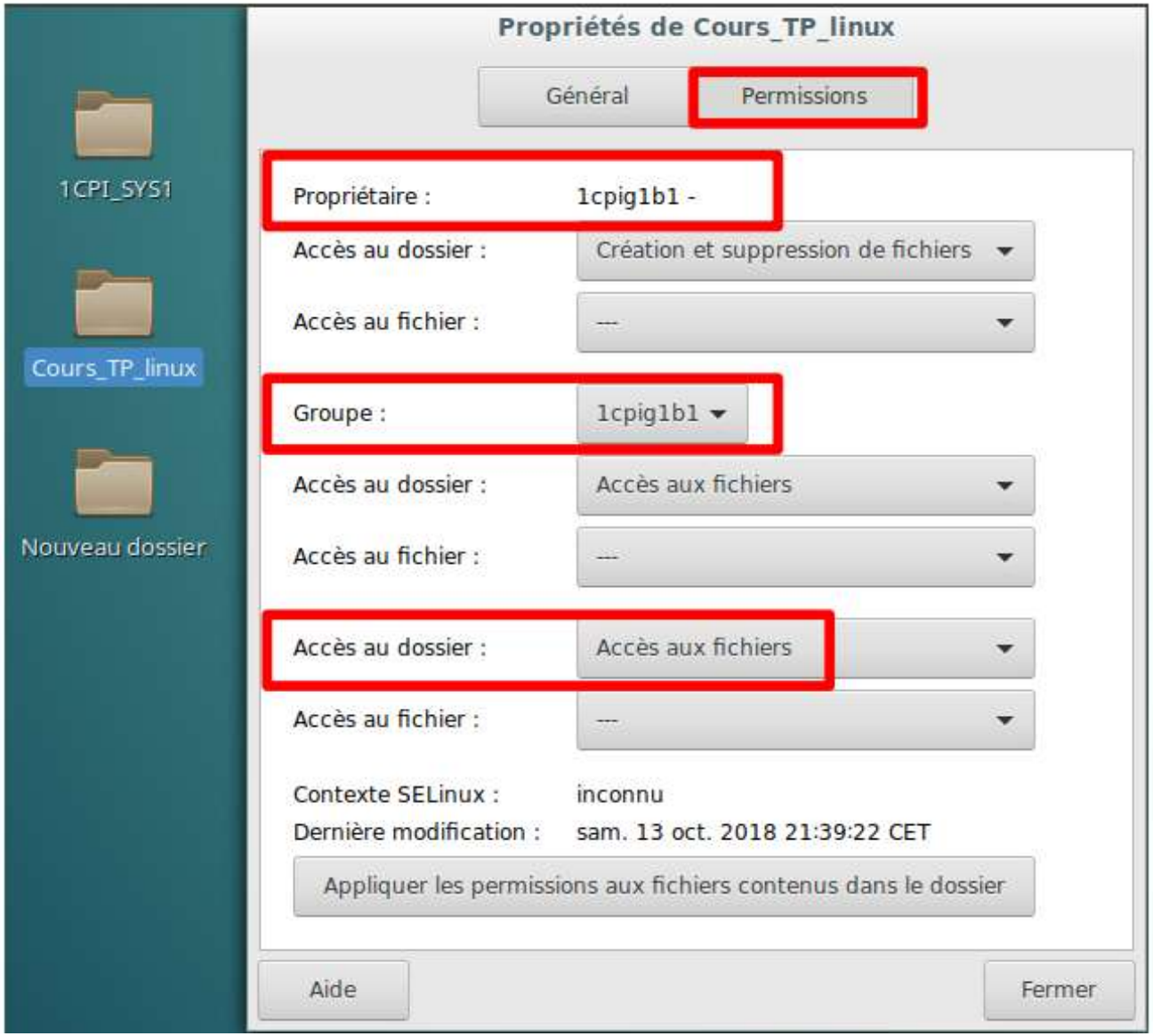
- **r = Read: On peut lire le contenu du répertoire**
- **w = Write :On peut modifier le contenu du répertoire c'est-à-dire il est possible d'ajouter, créer, supprimer ou renommer des fichers dans le répertoire**
- **x = Execute : On peut exécuter des commandes sur le contenu de répertoire**

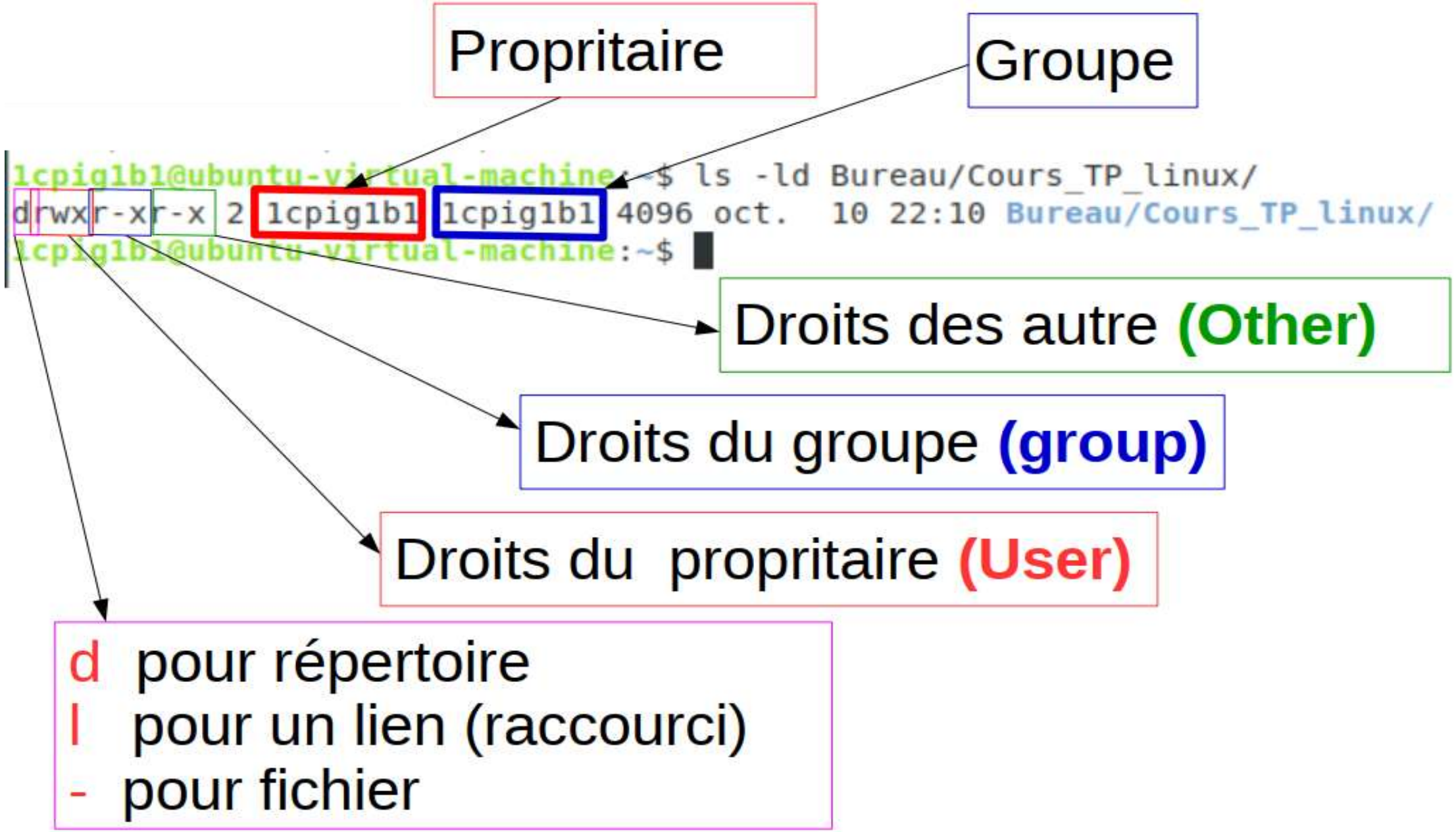
Signification des droits sur un répertoire

Attention:

Sans l'autorisation d'exécution (x), le répertoire devient verrouillé donc il est impossible d'agir sur son contenu c'est-à-dire ni accéder, ni ajouter, ni créer, ni supprimer ou renommer des fichiers ou des répertoires dans son contenu malgré le répertoire porte le droit d'écriteure (w)

Exemple par interface graphique




Propritaire
Groupe
lcpig1b1@ubuntu-virtual-machine:~$ ls -ld Bureau/Cours_TP_linux/
drwxr-xr-x 2 lcpig1b1 lcpig1b1 4096 oct. 10 22:10 Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$
Droits des autre (Other)
Droits du groupe (group)
Droits du propritaire (User)
d pour répertoire
l pour un lien (raccourci)
- pour fichier

Modification des droits

-En ligne de commande

$ chmod [options] modifications Fich/Rep ...

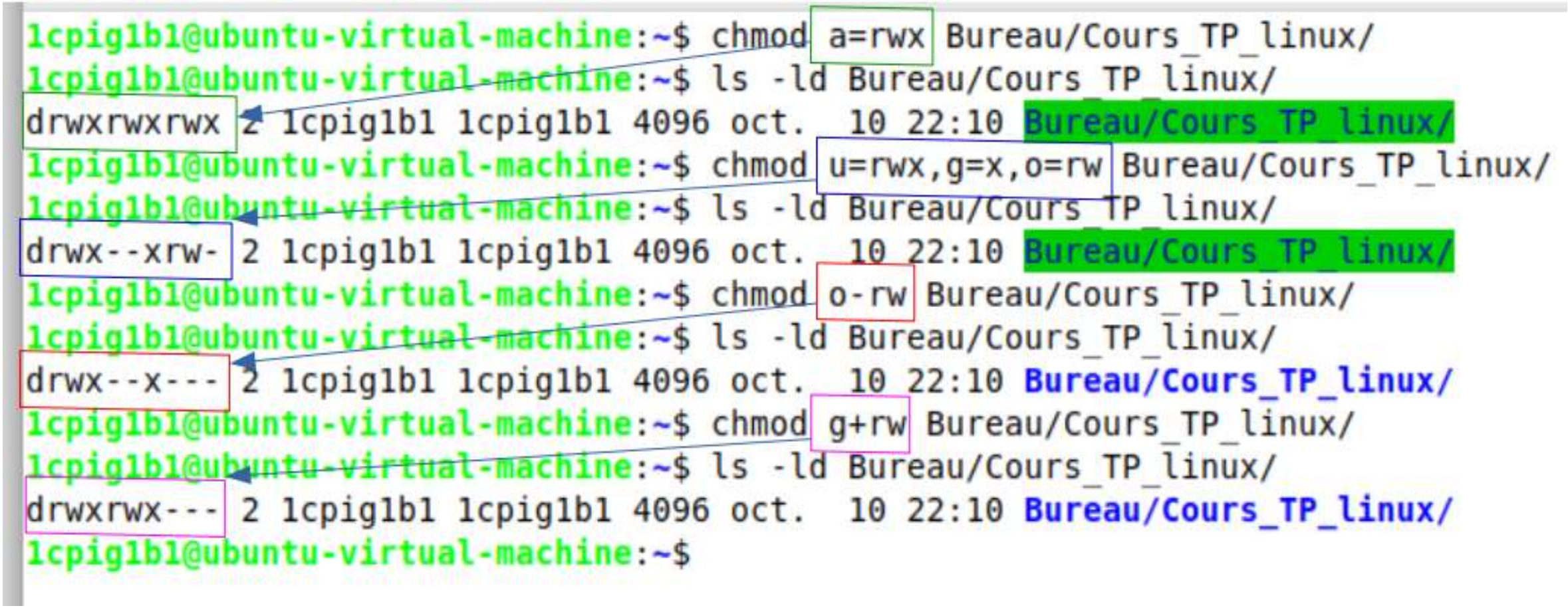
Par symboles

On utilise les caractères suivants

- **u** Pour modifier les droits de l'utilisateur
- **g** Pour modifier les droits de groupe
- **o** Pour modifier les droits des autres utilisateurs
- **a** Pour modifier les droits de l'utilisateur, groupe et les autres utilisateurs

- **+ Pour ajouter des droits**
- **- Pour retirer des droits**
- **= Pour affecter des droits**
- **r, w, x répresentent les droits**

Exemples:


lcpig1b1@ubuntu-virtual-machine:~$ chmod a=rwx Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ ls -ld Bureau/Cours_TP_linux/
drwxrwxrwx 2 lcpig1b1 lcpig1b1 4096 oct. 10 22:10 Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ chmod u=rwx,g=x,o=rw Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ ls -ld Bureau/Cours_TP_linux/
drwx--xrw- 2 lcpig1b1 lcpig1b1 4096 oct. 10 22:10 Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ chmod o-rw Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ ls -ld Bureau/Cours_TP_linux/
drwx--x--- 2 lcpig1b1 lcpig1b1 4096 oct. 10 22:10 Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ chmod g+rw Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$ ls -ld Bureau/Cours_TP_linux/
drwxrwx--- 2 lcpig1b1 lcpig1b1 4096 oct. 10 22:10 Bureau/Cours_TP_linux/
lcpig1b1@ubuntu-virtual-machine:~$

Modification des droits en ligne de commande (cont.)

Par des valeurs numériques en base 8 (octal)

- r est équivalent à la valeur numérique 4 (**r--=100**)
- w est équivalent à la valeur numérique 2 (**-w-=010**)
- x est équivalent à la valeur numérique 1 (**--x=001**)

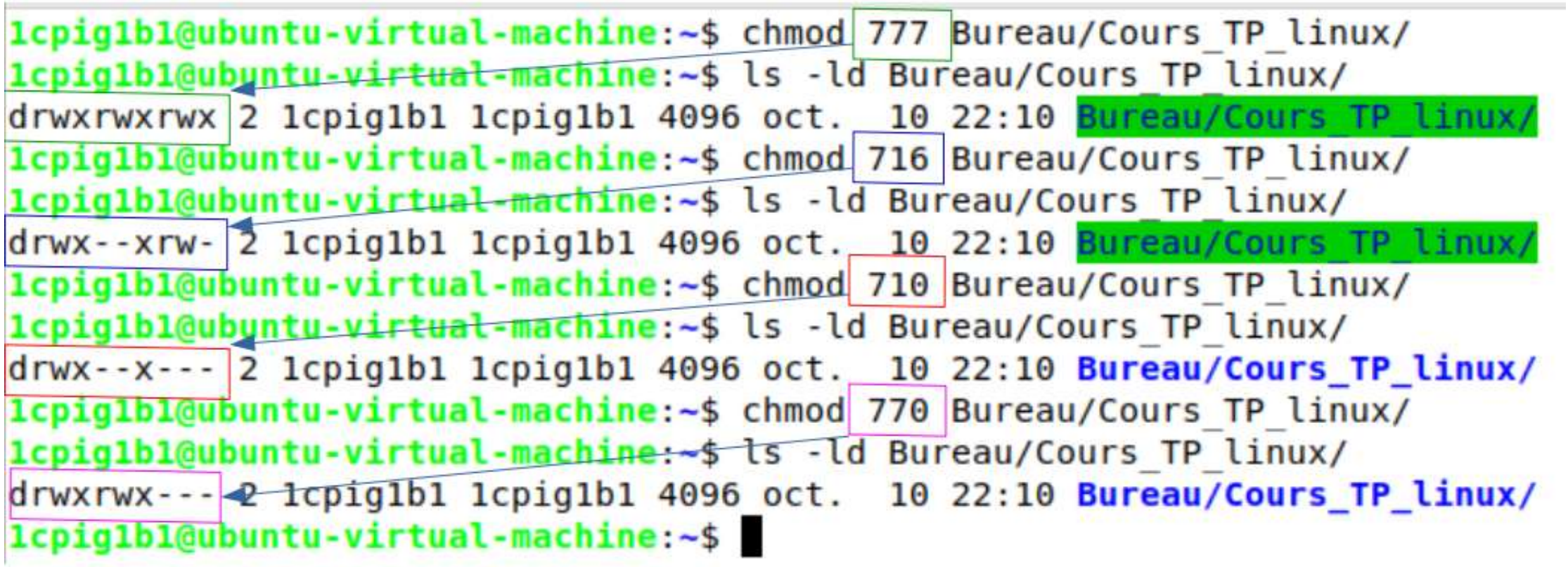
Donc:

rwx est équivalent à la valeur numérique **7=4+2+1**

rw- est équivalent à la valeur numérique **6=4+2+0**

r-x est équivalent à la valeur numérique **5=4+0+1**

Exemples de changement de mode d’acces par valeurs numériques



Droits attribués automatiquement aux fichiers et aux répertoires

Lorsqu'un nouveau fichier (ou répertoire) est créé, le SGF lui accorde des permissions d'accès par défaut.

- **Les droits par défaut attribué à un fichier :**

rw- rw- rw- (666) (les droits maximum)

- **Les droits par défaut attribué à un répertoire:**

rwx rwx rwx (777) (les droits maximum)

- **Ces permissions sont déterminées par un paramètre particulier appelé le masque utilisateur ou umask (user mask).**

(a) Calcule des droits pour un fichier

Par défaut	**rw- rw- rw-**	**(666)**
Masque	**000 010 010**	**(022)**
Reste	**r w- r - - r - -**	**(644)**

Donc pour le masque (022) les droits attribués aux fichiers automatiquement sont (644)

(b) Calcule des droits pour un répertoire

Par défaut	rwx rwx rwx	(777)
Masque	000 010 010	(022)
Reste	rwx r-x r - x	(755)

Donc pour le masque (022) les droits attribués aux fichiers automatiquement sont (755)

Synthèse:

On peut utiliser la formule suivant en **binaire** pour calculer les droits automatiques;

Les droits de fichier ou répertoire and ~~masque~~

Exemple dans le cas des fichiers:

Par défaut	**rw- rw- rw-**	**(666)**
Masque	**000 011 111**	**(037)**
~~Masque~~	**111 100 000**	**(740)**
Résultat	**rw- r-- ---**	**(640)**

Donc les droits automatiques pour le masque **(037)** sont **(640)**

La commande umask est utilisée pour contrôler la valeur de masque.

- **Pour savoir la valeur de masque en numérique**

$ umask

- **Pour savoir la valeur de masque en symbolique**

$ umask -S **NB. ‘S’ en majuscule**

- **Pour modifier le masque par valeur numérique**

$ umask [la valeur numériques]

- **Pour modifier le masque par symboles**

$ umask -S [la valeur en symboles]

Exemple 01:

On veux masquer en numérique automatiquement tous droits attribués à autre (other).

Question:

Quelle est la valeur du masque en numérique qu'on pourra utiliser ?

```
1cpig1b1@ubuntu-virtual-machine:~$ umask 007
1cpig1b1@ubuntu-virtual-machine:~$ umask
0007
1cpig1b1@ubuntu-virtual-machine:~$ mkdir Rep
1cpig1b1@ubuntu-virtual-machine:~$ ls -ld Rep/
drwxrwx--- 2 1cpig1b1 g1 4096 oct.  24 09:06 Rep/
1cpig1b1@ubuntu-virtual-machine:~$ touch Fich
1cpig1b1@ubuntu-virtual-machine:~$ ls -l Fich
-rw-rw--- 1 1cpig1b1 g1 0 oct.  24 09:07 Fich
1cpig1b1@ubuntu-virtual-machine:~$
```

Exemple 02:

On veux masquer en symblique automatiquement tous droits de groupe (group) et autre.

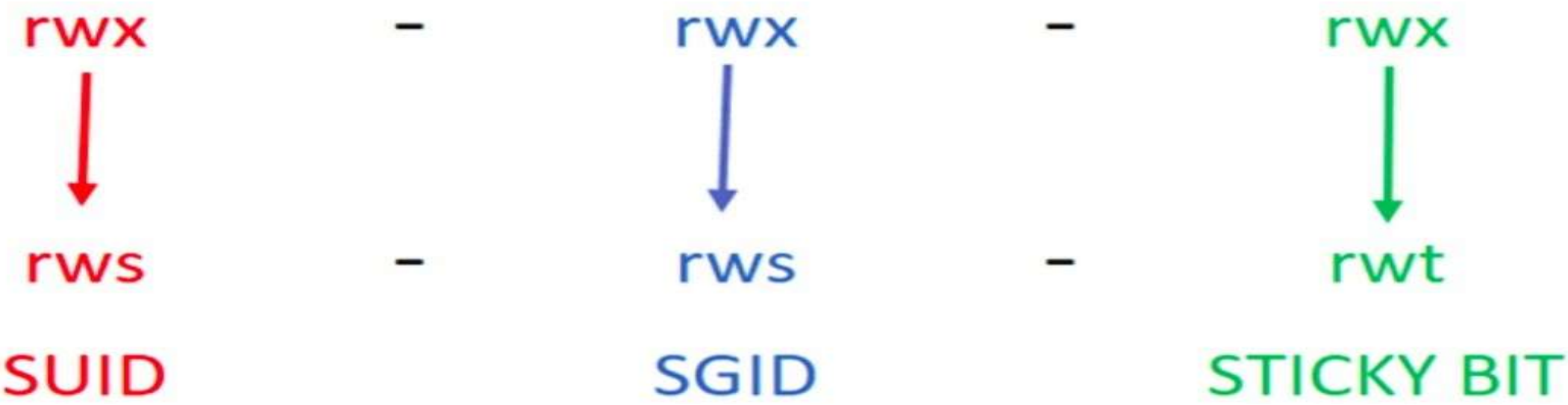
Question:

Quelle est la valeur du masque en symblique qu'on pourra utiliser ?

```
1cpig1b1@ubuntu-virtual-machine:~$ umask -S u=rwx,g=,o=
u=rwx,g=,o=
1cpig1b1@ubuntu-virtual-machine:~$ mkdir Rep2
1cpig1b1@ubuntu-virtual-machine:~$ ls -ld Rep2
drwx------ 2 1cpig1b1 g1 4096 oct.  24 09:22 Rep2
1cpig1b1@ubuntu-virtual-machine:~$ touch Fich2
1cpig1b1@ubuntu-virtual-machine:~$ ls -l Fich2
-rw------- 1 1cpig1b1 g1 0 oct.  24 09:23 Fich2
1cpig1b1@ubuntu-virtual-machine:~$
```

Les droits d’accès étendus :SUID, SGID et Sticky Bits

Il existe **3** droits **spéciaux** : **SUID, SGID et Sticky Bits** qui sont également attribuées avec la commande chmod. On les place devant les permissions comme suit:



Les droits d'accès étendus :SUID, SGID et Sticky Bits

Il arrive qu'un utilisateur doive accomplir une tache qui va au de la de ses droits : par exemple, pour changer votre propre mot de passe, vous devez modifier le fichier /etc/passwd — mais vous n'avez évidemment pas le droit d'écrire directement dans ce fichier.

Les droits d'accès étendus :SUID, SGID et Sticky Bits

Pour résoudre cela, Linux utilise des permissions supplémentaires, qui autorisent un programme à s'executer avec les droits du propriétaire du fichier, *root la plupart du temps*, plutot qu'avec ceux de l'utilisateur qui l'exécute. Ce système doit être utilisé de façon très prudente, car le programme executé heritera de toutes les permissions du propriétaire, et vous devrez etre certain qu'il n'accomplira que les taches pour lesquelles il est prevu. Par exemple, il ne faut pas que le programme **passwd** permette à un utilisateur de modifier le mot de passe de quelqu'un d'autre !

Les droits d'accès étendus :SUID, SGI et Sticky Bits

Cette permission peut s'appliquer au niveau du propriétaire ; on parle alors de bit **SUID (Set User ID)**, et le programme reçoit alors l'identité (et donc les privilèges) de son propriétaire lors de son execution. Il peut également s'appliquer au niveau du groupe, c'est alors le bit **SGID (Set Group ID)**, et le programme reçoit alors les privilèges de son groupe.

Au niveau des permissions, ces permissions se traduisent par un s qui remplace le x de l'autorisation d'exécution (si le programme ne possédait pas la permission x, la nouvelle permission est marquée par un S majuscule) :

:SUID

SUID (Set User ID): un droit qui s'applique uniquement a des fichiers.

Exemple:

Pour changer son mot de passe on utilise la commande **passwd**, mais pour que cela soit effectif en tant que user nous devons pouvoir écrire sur /etc/passwd et /etc/shadow alors qu'il ne nous est pas permis.

- **SUID** nous permet cette action.
- Allez voir les droits du fichier passwd avec la commande:

ls -l /usr/bin/passwd

:SUID

Notez ici qu'il y a un s à la place du x, donc nous avons les droits pour modifier un mot de passe car le script passwd a les autorisations SUID

:SUID

- Ce droit est accordé au script passwd par la command:

chmod u+s /usr/bin/passwd

- Il est possible aussi d’enlever ce droit avec la commande :

chmod u-s /usr/bin/passwd

:SUID

Donc:

Pour qu'un programme s'exécute avec les droits de propriétaire on ajoute le bit SUID au programme en utilisant la commande:

$ **chmod u+s programme ou commande**

SGID

- **SGID (Set Group ID):** s'applique aussi bien aux fichiers qu'aux répertoires.
- Pour les répertoires on ne parle pas de droits d'exécution mais d'appartenance.

- **Exemple**, si dans un répertoire appartenant à un groupe (avec droits SGID) plusieurs fichiers et/ou sous répertoires sont créés par un utilisateur, alors ces fichiers appartiendront à ce groupe d'utilisateurs.

:SGID

Comment peut-on accorder ces droits ?

```
fairouz@zxc-mnbvcxz:~$ chmod g+s ISE1
fairouz@zxc-mnbvcxz:~$ ls -ld ISE1
drwxr-sr-x 5 fairouz fairouz 4096 بفون 7 11:00 ISE1
fairouz@zxc-mnbvcxz:~$
```

- **Conclusion:**

  SUID et SGID sont de puissantes autorisations spéciales pour les exécutables et les répertoires sous Linux.

Les droits d'accès étendus : Sticky Bits

- **Partage d'un répertoire avec d'autres (others) utilisateurs du système: Sticky Bits**

- **Exemple:** un enseignant user1 veux partager un répertoire rep avec ses étudiants user2 et user3 pour qu'ils puissent mettre leurs fichiers: TP_user2.docx et TP_user3.docx

- **Solution 1:**
  user1 accorde à user2 et user3 (others) les droits rwx

```
user1@_    PC    :~$ chmod o+rw rep/
user1@_    PC    :~$ ls -ld rep/
drwxrwxrwx 2 user1 user1 4096          11:43 rep/
```

Les droits d'accès étendus : Sticky Bits

- **Inconvénient:** un utilisateur (exp. user3) peut supprimer le fichier de l'autre (user2).

```
user2@     PC    :~$ cp TP_user2.docx /home/user1/rep/
user2@     PC    :~$ su user3
Mot de passe :
user3@     PC    :/home/user2$ cd
user3@     PC    :~$ cp TP_user3.docx /home/user1/rep/
user3@     PC    :~$ ls -l /home/user1/rep/
total 0
-rw-rw-r-- 1 user2 user2 0                11:55 TP_user2.docx
-rw-rw-r-- 1 user3 user3 0                11:55 TP_user3.docx
user3@     PC    :~$ rm -rf /home/user1/rep/*
user3@     PC    :~$ ls -l /home/user1/rep/
total 0
```

Les droits d'accès étendus : Sticky Bits

- Il est possible d'établir des droits d'accès étendus sur un répertoire partagé afin d'empêcher un utilisateur de supprimer les fichiers des autres malgré que le répertoire porte les droits rwx pour autre (o).

  - Pour cela il suffit d'ajouter **le Sticky bit** en utilisant à l'aide de la commande:

$ chmod o+t répertoire

Les droits d'accès étendus :Sticky Bits

```
user1@   PC   :~$ chmod o+t rep/
user1@   PC   :~$ ls -ld rep/
drwxrwxrwt 2 user1 user1 4096          11:56 rep/
user1@   PC   :~$ su user2
Mot de passe :
user2@   PC   :/home/user1$ cd
user2@   PC   :~$ cp TP_user2.docx /home/user1/rep/
user2@   PC   :~$ su user3
Mot de passe :
user3@   PC   :/home/user2$ cd
user3@   PC   :~$ cp TP_user3.docx /home/user1/rep/
user3@   PC   :~$ rm -rf /home/user1/rep/TP_user2.docx
rm: impossible de supprimer '/home/user1/rep/TP_user2.docx': Opération non permise
user3@   PC   :~$ ls -l /home/user1/rep/
total 0
-rw-rw-r-- 1 user2 user2 0                    TP_user2.docx
-rw-rw-r-- 1 user3 user3 0                    TP_user3.docx
```

Les droits d'accès étendus :SUID, SGID et Sticky Bits

➔ **Les droits d'accès étendus en numérique:**

- la valeur **4 (100)** pour ajouter le bit **SUID**
- la valeur **2 (010)** pour ajouter le bit **SGID**
- la valeur **1 (001)** pour ajouter le **Sticky bit**

➔ **Exemple:**

```
user1@    PC    :~$ chmod 4755 fichier1
user1@    PC    :~$ chmod 2755 fichier2
user1@    PC    :~$ chmod 1755 repertoire/
user1@    PC    :~$ ls -l fichier1 fichier2
-rwsr-xr-x 1 user1 user1 0          19:22 fichier1
-rwxr-sr-x 1 user1 user1 0          19:22 fichier2
user1@    PC    :~$ ls -ld repertoire/
drwxr-xr-t 2 user1 user1 4096       19:22 repertoire/
```