

Ecole Nationale des Sciences Appliquées Khouribga

UNIX GNU/Linux Droits d'accès

Med AMNAI
2017–2018

Plan

1- Droits d'accès

2- Modification des droits d'accès

1. Méthode symbolique

2. Méthode octal

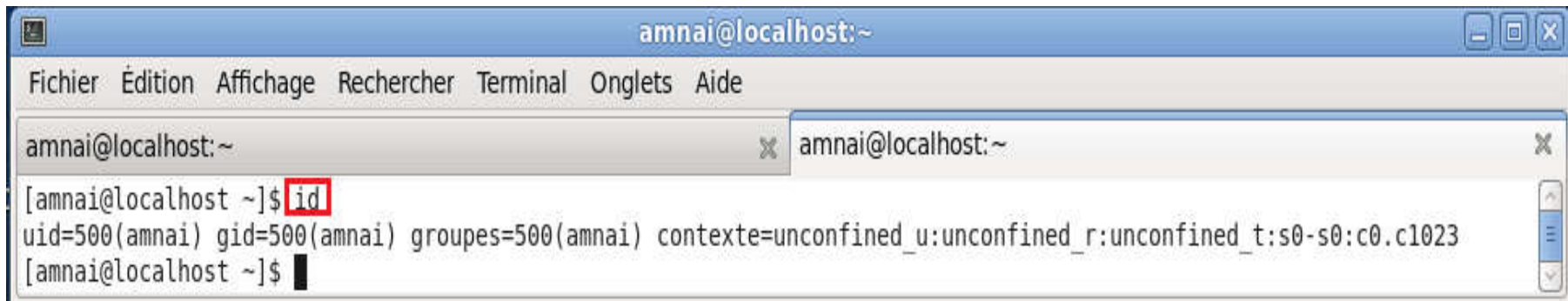
3 - Modification du propriétaire d'un fichier/répertoire

4 - Modification du groupe d'un fichier/répertoire

5 - Enlever des permissions par défaut pour les nouveaux fichiers

1- Droits d'accès

- ❑ Linux est un système multi-utilisateur qui permet de gérer les permissions d'accès aux fichiers.
- ❑ Chaque utilisateur a un identifiant (**UID**), un nombre unique qui l'identifie.
- ❑ Les utilisateurs appartiennent également à un ou plusieurs groupes.
- ❑ **Les groupes peuvent être employés pour limiter l'accès à un certain nombre de personnes.**
- ❑ Pour vérifier votre identification d'utilisateur et voir le groupe(s) auquel vous appartenez, tapez la commande **id**:



The screenshot shows a terminal window titled 'amnai@localhost:~'. The window has a menu bar with 'Fichier', 'Édition', 'Affichage', 'Rechercher', 'Terminal', 'Onglets', and 'Aide'. Below the menu bar, there are two tabs, both labeled 'amnai@localhost:~'. The terminal content shows the command '[amnai@localhost ~]\$ id' being entered, with 'id' highlighted by a red box. The output of the command is 'uid=500(amnai) gid=500(amnai) groupes=500(amnai) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023'. The prompt '[amnai@localhost ~]\$' is followed by a cursor.

```
amnai@localhost:~  
Fichier Édition Affichage Rechercher Terminal Onglets Aide  
amnai@localhost:~ x amnai@localhost:~ x  
[amnai@localhost ~]$ id  
uid=500(amnai) gid=500(amnai) groupes=500(amnai) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[amnai@localhost ~]$
```

1- Droits d'accès : Propriété et permissions des fichiers

- ❑ Les opérations qui peuvent être effectuées sur un fichier sont: la **lecture**, **l'écriture** et **l'exécution**.
- ❑ Mais a-t-on pour autant le droit de lire, de modifier ou d'exécuter n'importe quel fichier ?
- ❑ Il existe un mécanisme permettant de protéger les fichiers des utilisateurs contre l'indiscrétion et la malveillance d'autrui.
- ❑ Il repose sur les **permissions** des **fichiers**, qui sont un des concepts de base de la gestion d'utilisateurs multiples sous Unix.
- ❑ Il va permettre **d'interdire/autoriser** la **lecture**, **l'écriture** ou **l'exécution** de certains fichiers par certains utilisateurs.

1- Droits d'accès : Principe

- ❑ À chaque fois qu'un utilisateur veut effectuer une opération sur un fichier, le système va vérifier que cette opération lui est permise.
- ❑ Cette vérification repose sur les 5 informations suivantes :
 - **l'utilisateur** qui tente d'effectuer l'opération ;
 - ses **groupes** ;
 - le **propriétaire** du fichier ;
 - le **groupe** du fichier ;
 - les **permissions** (ou droits d'accès) du fichier.
- ❑ Les trois dernières informations, relatives au fichier, peuvent être visualisées avec la commande **ls** (option **-l**).

1- Droits d'accès : Principe

- ❑ Les permissions du fichier sont composées de trois parties, chacune s'adressant à une catégorie d'utilisateur :
 - les droits du **propriétaire** ;
 - les droits des membres du **groupe** du fichier ;
 - les droits des **autres utilisateurs**.
- ❑ Les droits d'un utilisateur sur un fichier sont uniquement ceux de sa catégorie la plus spécifique (**propriétaire**, sinon **membre du groupe**, sinon **autre**).
- ❑ L'opération demandée par l'utilisateur ne sera autorisée que s'il possède les droits qu'elle nécessite.

1- Droits d'accès : **fichier**

- ☐ **Lecture** pour en **lire** le contenu ;
- ☐ **Écriture** pour en **modifier** le contenu ;
- ☐ **Exécution** pour **l'exécuter** s'il s'agit d'un **fichier binaire** contenant du code exécutable. S'il s'agit d'un fichier **texte** (contenant un script), **il faut aussi le droit de lecture** pour **l'exécuter**

1- Droits d'accès : **répertoire**

- ❑ **Lecture** consulter son contenu (afficher la liste des fichiers qu'il contient), notamment par la commande **ls** ;
- ❑ **Écriture** modifier le répertoire, ce qui comprend **l'ajout** d'un fichier (de tout type), sa **suppression** (même sans droit sur ce fichier !) ou son **renommage** (sans changer de propriétaire).

RQ : Le droit **d'écriture** ne sert à rien si on n'a pas aussi le droit **d'exécution** ;

- ❑ **Exécution** autoriser **l'accès** au répertoire. Son absence permet donc d'interdire l'accès à une partie de l'arborescence du système de fichiers.

1- Droits d'accès : **répertoire**

En effet, sans le droit **d'exécution** sur un répertoire, un utilisateur ne peut pas le "traverser" et ne peut pas :

- ☐ faire de ce répertoire son **répertoire** de travail, c'est à dire aller dans ce répertoire (en utilisant **cd**) ;
- ☐ **référencer** un fichier qui se trouve dans l'arborescence de ce répertoire ;
- ☐ **obtenir** des **détails** sur les fichiers contenus dans ce répertoire (notamment avec l'option **-l de ls**).

1- Droits d'accès : **Connaître ses droits sur un fichier**

Les permissions d'accès pour un fichier peuvent être positionnés par **propriétaire, groupe** et pour **les autres** sur la base de permissions en lecture **(r)**, écriture **(w)** et exécution **(x)**. Les droits d'accès sont attribués aux fichiers ou aux répertoires.

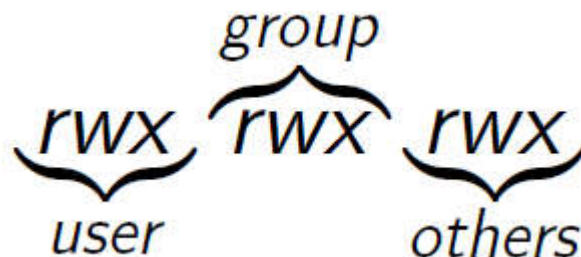
Il y a trois catégories de droits :

- Lecture (**read**).
- Écriture (**write**).
- Exécution (**execute**).

1- Droits d'accès : Représentation

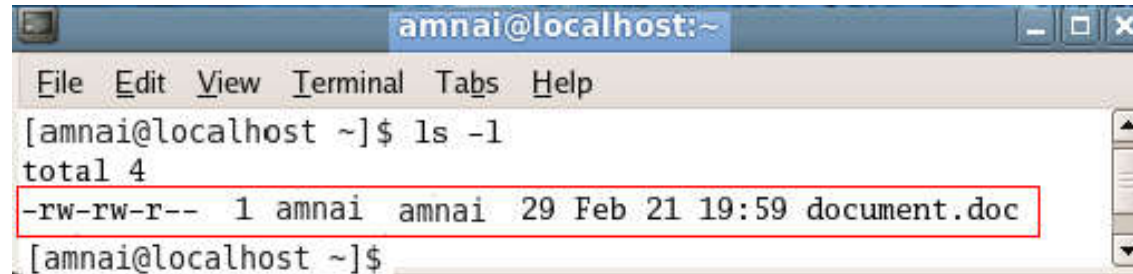
Les droits sont définis pour trois types d'utilisateurs:

- le propriétaire du fichier (**u=user**)
- le groupe auquel appartient le fichier (**g=group**)
- tous les autres utilisateurs (**o=other**)
- **a=all** signifie **user+group+other**



1- Droits d'accès : exemple

Exemple:

A terminal window titled 'amnai@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command '[amnai@localhost ~]\$ ls -l' has been executed. The output shows 'total 4' followed by a line for 'document.doc' with permissions '-rw-rw-r--', owner 'amnai', group 'amnai', and date '29 Feb 21 19:59'. This line is highlighted with a red rectangle. The prompt '[amnai@localhost ~]\$' is visible at the bottom.

```
amnai@localhost:~  
File Edit View Terminal Tabs Help  
[amnai@localhost ~]$ ls -l  
total 4  
-rw-rw-r-- 1 amnai amnai 29 Feb 21 19:59 document.doc  
[amnai@localhost ~]$
```

- Le premier tiret signifie que c'est un fichier classique (**d pour un répertoire**).
- Les trois caractères suivants (**rw-**) montrent les droits de l'utilisateur **propriétaire** du fichier.
- Les trois caractères suivants (**rw-**) montrent les droits du **groupe** auquel appartient le fichier.
- Les trois derniers caractères (**r--**) montre les droits des **autres utilisateurs**.
- **1** est le nombre de **liens** vers le fichier concerné.
- **amnai** est le **propriétaire** du fichier.
- **amnai** est le **groupe** auquel ce fichier appartient.

2-Modification des droits d'accès

- ❑ Les droits d'accès ne peuvent être **modifiés** que par le **propriétaire** ou **l'administrateur**.
- ❑ La commande **chmod** permet de modifier les droits d'accès. Elle peut être utilisée de deux façons différentes (**symbolique** et **octal**).
- ❑ Pour pouvoir changer les droits on doit spécifier:
 - Les **droits** (r=read, w=write, x=execute).
 - A **qui** s'appliquent ces droits (u=user, g=group, o=other, a=all).
 - Le ou les fichiers/répertoires dont on veut changer les droits.

2-Modification des droits d'accès :

Mode symbolique

- ❑ Dans ce mode, les permissions ont la forme suivante (les parenthèses ne servent ici qu'à la lisibilité) :

Chmod *qui* *opérateur* *quoi* {,*qui* *opérateur* *quoi*}

- ❑ où ***qui*** indique à qui on veut fixer des permissions. C'est une combinaison des lettres **u**, **g**, **o** et **a**, représentant :
 - **u (user)** le propriétaire ;
 - **g (group)** les membres du groupe ;
 - **o (others)** les autres ;
 - **a (all)** à la fois le propriétaire, les membres du groupe et les autres.
- ❑ ***L'opérateur*** est l'un des signes **+**, **=** et **-**, pour indiquer si l'on veut :
 - **+** ajouter des permissions,
 - **=** fixer exactement des permissions,
 - **-** supprimer des permissions.

2-Modification des droits d'accès

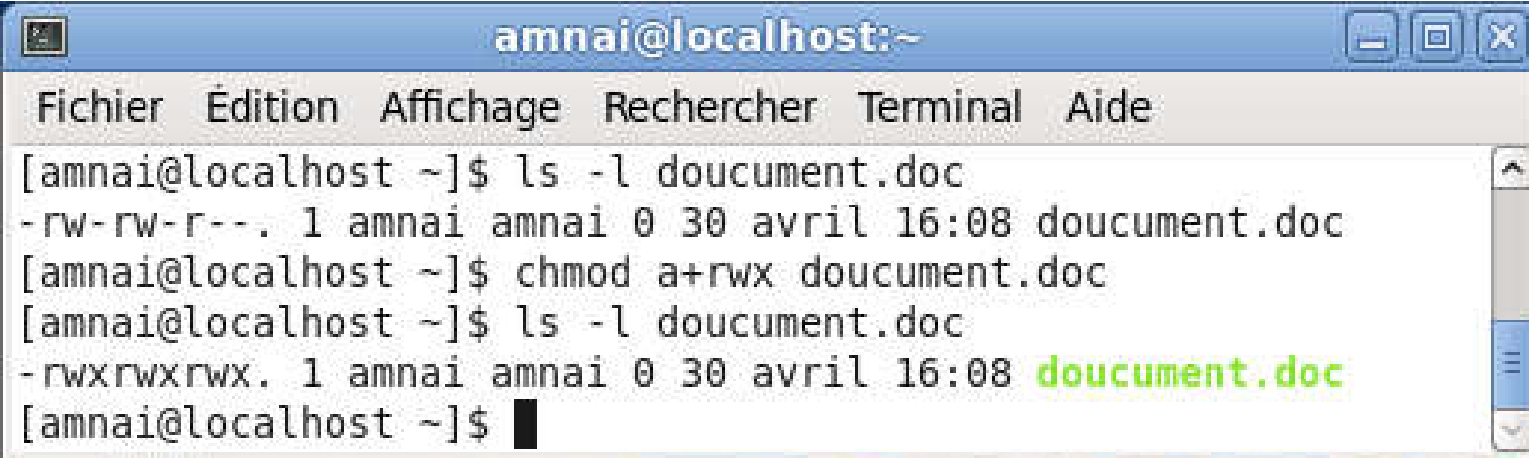
1-Méthode symbolique

Dans l'exemple suivant on donne (+) tous les droits (**rw****x**) d'accès à tous les utilisateurs (**a** ou **ugo**) :

\$chmod a+rw document.doc

Ou

\$chmod ugo+rw document.doc



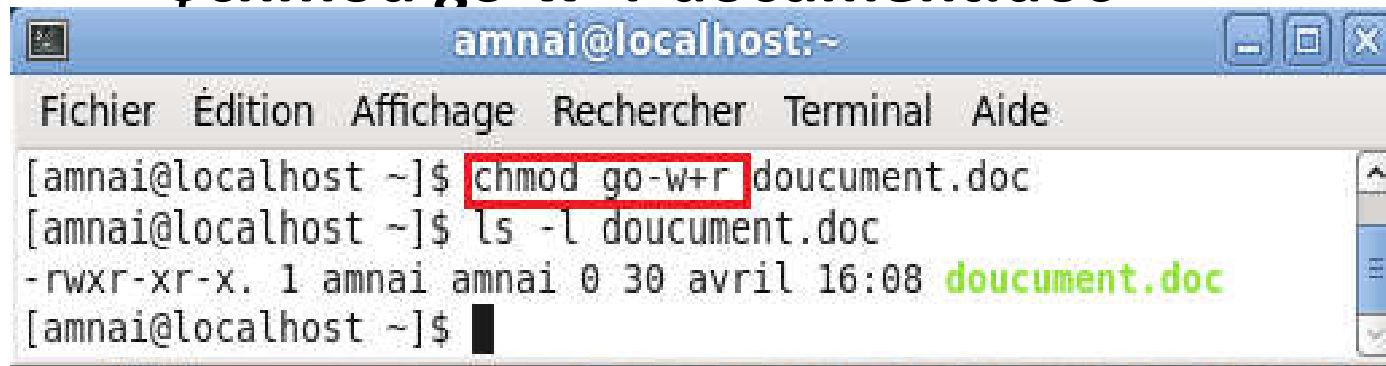
```
amnai@localhost:~  
Fichier  Edition  Affichage  Rechercher  Terminal  Aide  
[amnai@localhost ~]$ ls -l document.doc  
-rw-rw-r--. 1 amnai amnai 0 30 avril 16:08 document.doc  
[amnai@localhost ~]$ chmod a+rw document.doc  
[amnai@localhost ~]$ ls -l document.doc  
-rwxrwxrwx. 1 amnai amnai 0 30 avril 16:08 document.doc  
[amnai@localhost ~]$
```

2-Modification des droits d'accès

1-Méthode symbolique

Dans l'exemple qui suit on enlève (-) le droit en écriture (**w**) et on rajoute (+) le droit en lecture (**r**) à **group (g)** et **other (o)**:

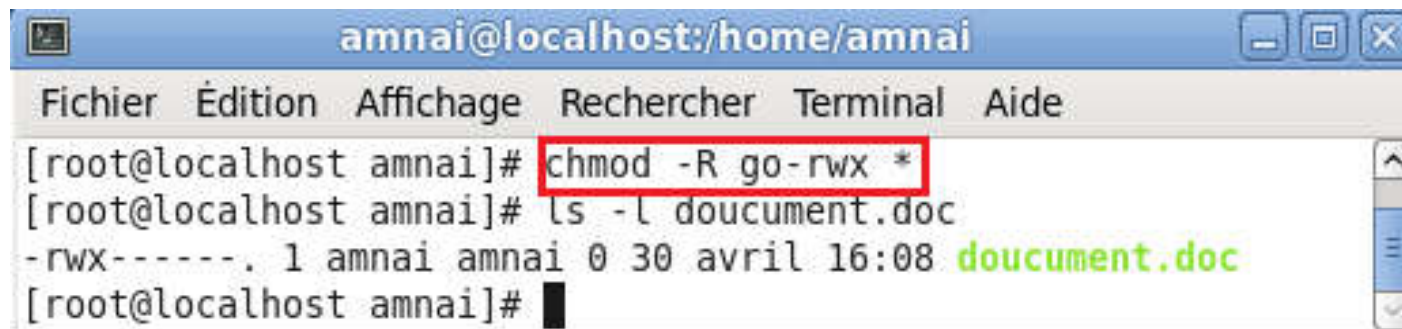
\$chmod go-w+r document.doc



```
amnai@localhost:~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
[amnai@localhost ~]$ chmod go-w+r document.doc  
[amnai@localhost ~]$ ls -l document.doc  
-rwxr-xr-x. 1 amnai amnai 0 30 avril 16:08 document.doc  
[amnai@localhost ~]$
```

Dans l'exemple qui suit on enlève le droit **read, write et execute** à **group et other** pour les fichiers du répertoire courant et de ses sous répertoires:

\$chmod -R go-rwx *



```
amnai@localhost:/home/amnai  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
[root@localhost amnai]# chmod -R go-rwx *  
[root@localhost amnai]# ls -l document.doc  
-rwx-----. 1 amnai amnai 0 30 avril 16:08 document.doc  
[root@localhost amnai]#
```


2-Modification des droits d'accès

1-Méthode octal

Dans ce mode, les permissions seront représentées en octal sous la forme $C_u C_g C_o$ où C_u , C_g et C_o sont 3 chiffres entre **0** et **7** :

C_u : indique les permissions du **propriétaire** ;

C_g : indique les permissions du **groupe** ;

C_o : indique les permissions des **autres**.

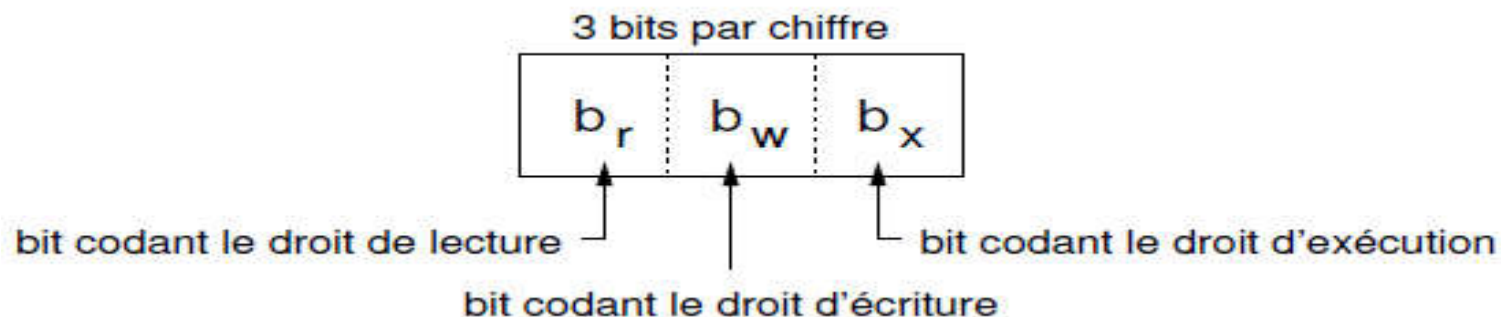
Chaque chiffre est obtenu en sommant les valeurs correspondant aux différents droits :

4 pour la lecture ;

2 pour l'écriture ;

1 pour l'exécution.

Plus exactement, chaque chiffre est un champ de 3 bits :



2-Modification des droits d'accès

1-Méthode octal (suite)

Dans l'exemple suivant on **supprime tous** les **droits** à tout le monde :

\$chmod 000 document.doc



```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
-rwxrwxrwx. 1 amnai amnai 0 30 avril 16:08 document.doc
[root@localhost amnai]# chmod 000 document.doc
[root@localhost amnai]# ls -l document.doc
-----. 1 amnai amnai 0 30 avril 16:08 document.doc
[root@localhost amnai]#
```

Dans l'exemple suivant on donne le droit de **lecture** et **d'écriture** ($6=4+2$) au **propriétaire** (664) et au **groupe** (664) et uniquement le droit de **lecture** (4) aux **autres** (664) utilisateurs :



```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
-----. 1 amnai amnai 0 30 avril 16:08 document.doc
[root@localhost amnai]# chmod 664 document.doc
[root@localhost amnai]# ls -l document.doc
-rw-rw-r--. 1 amnai amnai 0 30 avril 16:08 document.doc
[root@localhost amnai]#
```

2-Modification des droits d'accès

1-Méthode octal (suite)

L'option **-v** permet d'afficher les **modifications réalisées** :

664 -> (rw-rw-r--)



```
amnai@localhost:/home/amnai
Fichier Édition Affichage Rechercher Terminal Aide
[root@localhost amnai]# chmod -v 664 document.doc
le mode de « document.doc » a été conservé à 0664 (rw-rw-r--).
```

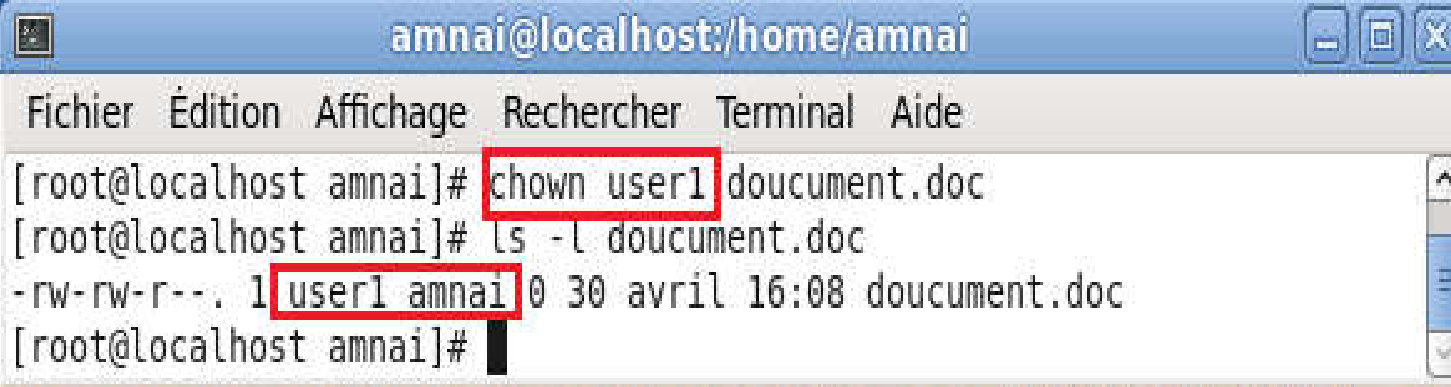
3-Modification du propriétaire d'un fichier/répertoire

- ❑ Seulement **root** peut changer le propriétaire (owner) d'un fichier ou répertoire.
- ❑ Pour modifier le propriétaire, vous pouvez utiliser les commandes **chown** (change owner):

\$chown user1 document.doc

ou

\$chown user1 /home/amnai/document.doc



The screenshot shows a terminal window titled 'amnai@localhost:/home/amnai'. The terminal output is as follows:

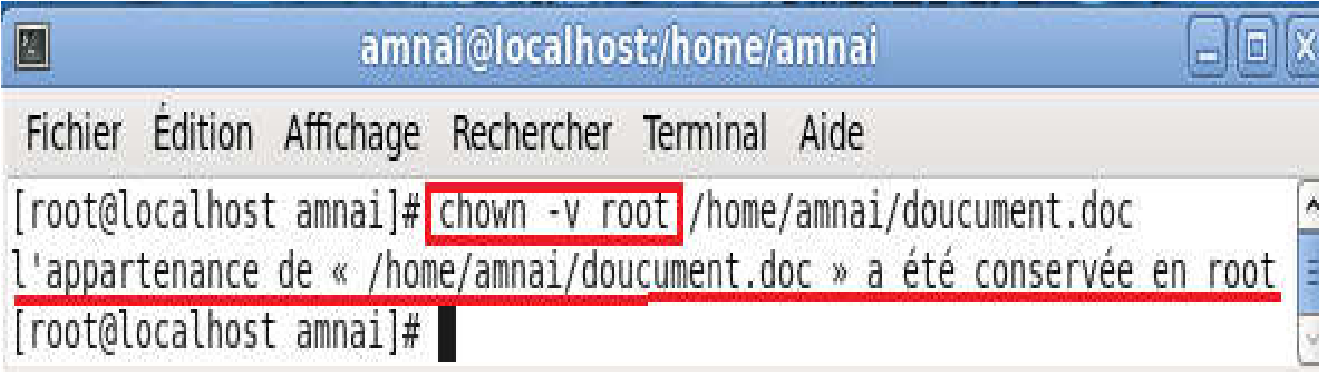
```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost amnai]# chown user1 document.doc
[root@localhost amnai]# ls -l document.doc
-rw-rw-r--. 1 user1 amnai 0 30 avril 16:08 document.doc
[root@localhost amnai]#
```

In the terminal output, the command 'chown user1' and the file permissions '1 user1 amnai' in the 'ls' output are highlighted with red boxes.

Dans cet exemple, **user1** devient le nouveau propriétaire du fichier **document.doc**.

3-Modification du propriétaire d'un fichier/répertoire

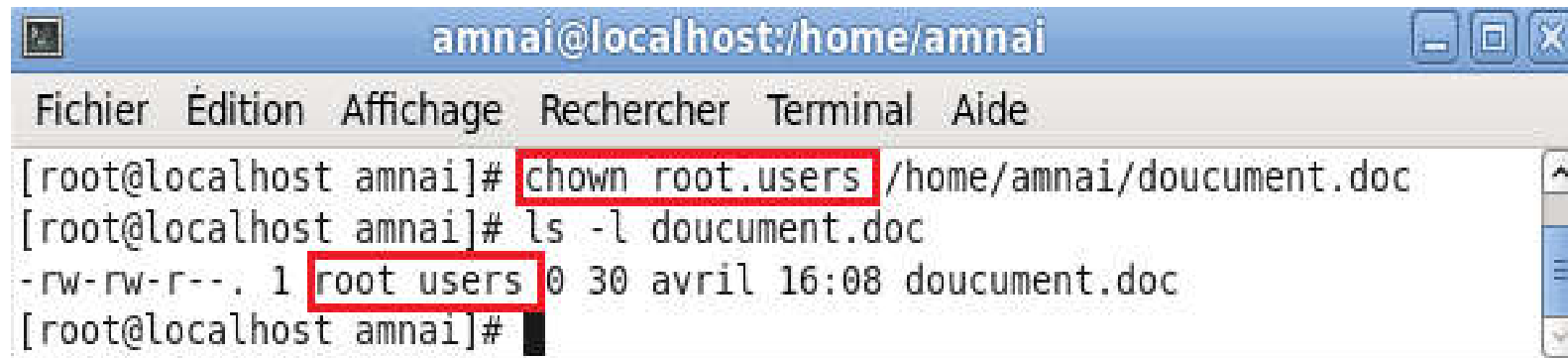
L'option **-v** permet **d'afficher** les **modifications** réalisées :



```
amnai@localhost:/home/amnai
Fichier Édition Affichage Rechercher Terminal Aide
[root@localhost amnai]# chown -v root /home/amnai/doucument.doc
l'appartenance de « /home/amnai/doucument.doc » a été conservée en root
[root@localhost amnai]#
```

3-Modification du propriétaire d'un fichier/répertoire

Pour changer le **propriétaire (owner) et le groupe** :



```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost amnai]# chown root.users /home/amnai/doucement.doc
[root@localhost amnai]# ls -l doucement.doc
-rw-rw-r--. 1 root users 0 30 avril 16:08 doucement.doc
[root@localhost amnai]#
```

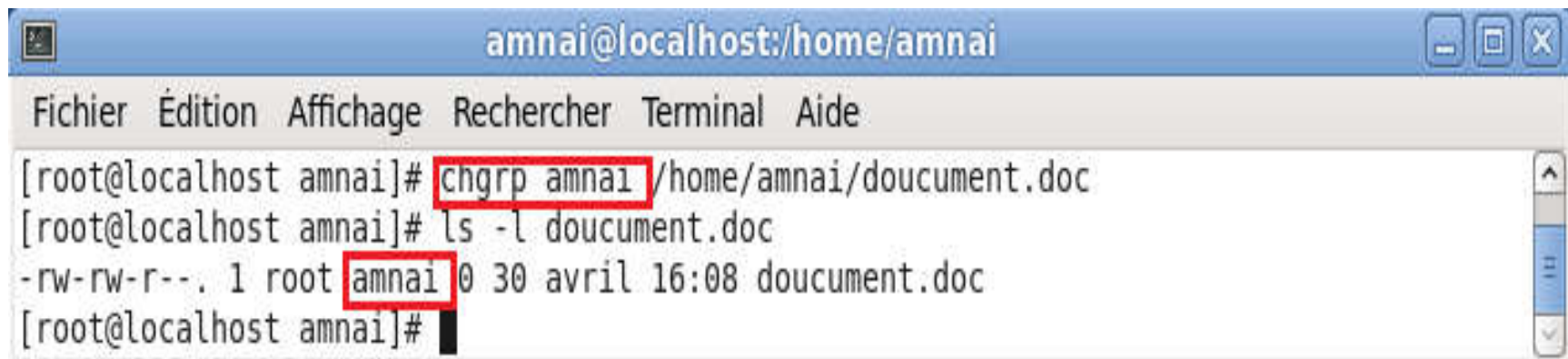
Pour savoir quels sont les **groupes où vous êtes membre**, tapez la commande **groups** :



```
amnai@localhost:~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost amnai]# groups
root bin daemon sys adm disk wheel
[root@localhost amnai]# su - amnai
[amnai@localhost ~]$ groups
amnai
[amnai@localhost ~]$
```

4-Modification du groupe d'un fichier/répertoire

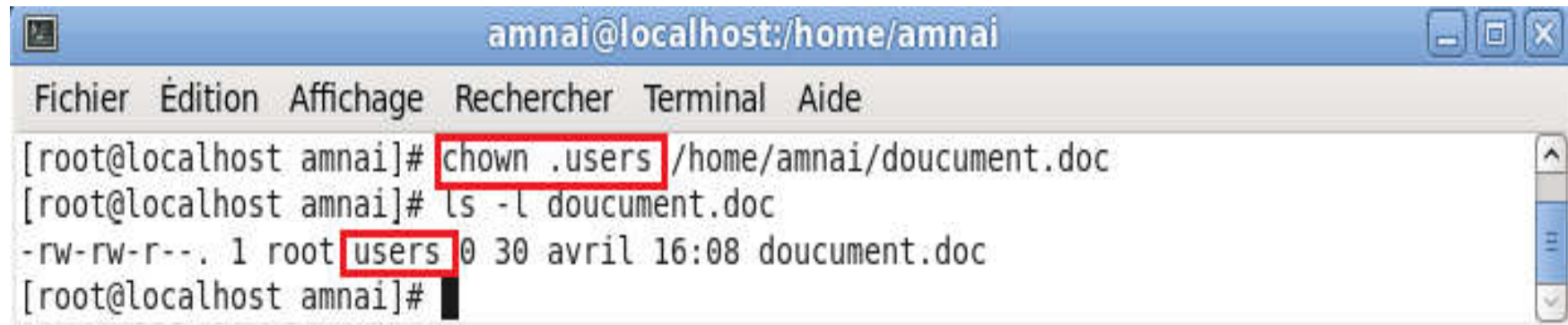
- ❑ Pour **modifier le groupe d'un fichier**, vous pouvez utiliser les commandes **chgrp** (change group).
- ❑ Seulement **root** et le **propriétaire (owner)** s'il appartient au nouveau groupe ont le droit de changer le groupe.
- ❑ Dans l'exemple suivant **amnai** devient le nouveau groupe du fichier **document.doc** :



```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost amnai]# chgrp amnai /home/amnai/doucement.doc
[root@localhost amnai]# ls -l doucement.doc
-rw-rw-r--. 1 root amnai 0 30 avril 16:08 doucement.doc
[root@localhost amnai]#
```

4-Modification du groupe d'un fichier/répertoire

On peut aussi utiliser la commande **chown**. Dans l'exemple suivant **users** devient le nouveau groupe du fichier **document.doc** :



```
amnai@localhost:/home/amnai
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost amnai]# chown .users /home/amnai/doucument.doc
[root@localhost amnai]# ls -l doucument.doc
-rw-rw-r--. 1 root users 0 30 avril 16:08 doucument.doc
[root@localhost amnai]#
```


5- umask : enlever les permissions par défaut pour les nouveaux fichiers

- ☐ Tous les fichiers ont des permissions fixées dès leur création. Elles dépendent de l'**utilitaire** employé et du **type de fichier créé**, ainsi que du **masque de création** de fichiers.
- ☐ Le **masque** indique au système les permissions que ne doivent pas avoir les fichiers lors de leur création (uniquement).
- ☐ Sous Bash, il est **consultable/modifiable** par la commande interne **umask**.

5- umask : enlever les permissions par défaut pour les nouveaux fichiers

☐ Syntaxe

umask [*masque*]

- ☐ Sans argument, **umask** indique la **valeur actuelle** du masque de création de fichiers, sinon masque est sa nouvelle valeur.
- ☐ L'argument masque est un nombre (en octal) de la forme $C_u C_g C_o$ où C_u , C_g et C_o sont 3 chiffres entre **0** et **7** :
 - C_u indique les permissions à **ne pas accorder** au propriétaire ;
 - C_g indique les permissions à **ne pas accorder** au groupe ;
 - C_o indique les permissions à **ne pas accorder** aux autres.

5- **umask** : enlever les permissions par défaut pour les nouveaux fichiers

```
[amnai@localhost ~]$ umask  
0002  
[amnai@localhost ~]$ mkdir tpunix  
[amnai@localhost ~]$ cd tpunix  
[amnai@localhost tpunix]$ touch fich1  
[amnai@localhost tpunix]$  
[amnai@localhost tpunix]$ ls -l  
total 0  
-rw-rw-r--. 1 amnai amnai 0 27 oct. 12:46 fich1  
[amnai@localhost tpunix]$  
[amnai@localhost tpunix]$ mkdir rep  
[amnai@localhost tpunix]$  
[amnai@localhost tpunix]$ ls -l  
total 4  
-rw-rw-r--. 1 amnai amnai 0 27 oct. 12:46 fich1  
drwxrwxr-x. 2 amnai amnai 4096 27 oct. 12:46 rep  
[amnai@localhost tpunix]$
```

Conformément au masque (002), le **droit d'écriture ne sera pas accordé aux autres.**

5- **umask** : enlever des permissions par défaut pour les nouveaux fichiers

```
[amnai@localhost tpunix]$ umask 267
[amnai@localhost tpunix]$
[amnai@localhost tpunix]$ touch fich2
[amnai@localhost tpunix]$ mkdir rep2
[amnai@localhost tpunix]$
[amnai@localhost tpunix]$ ls -l
total 8
-rw-rw-r--. 1 amnai amnai    0 27 oct.  12:46 fich1
-r-----. 1 amnai amnai    0 27 oct.  13:12 fich2
drwxrwxr-x. 2 amnai amnai 4096 27 oct.  12:46 rep
dr-x--x---. 2 amnai amnai 4096 27 oct.  13:12 rep2
[amnai@localhost tpunix]$
```

On ne veut pas accordés le droit **d'écriture (2=w)** pour le **propriétaire (267)**, les droits de **lecture et d'écriture (6=4+2)** pour le **groupe (267)** et l'ensemble des **droits (7=4+2+1)** pour les **autres (267)**.

Rq : L'affichage du **masque 000** sous forme **symbolique** en utilisant l'option **-S** de **umask**

\$ umask -S

u=rwx,g=rwx,o=rwx