***NOLLE***

***Damien***

***L3 Informatique***

**SR – Devoir 1 :**

|  |  |
| --- | --- |
| ***Note :*** | ***Observation :*** |
| ***/20*** |  |

Exercice 1)

Q1 :

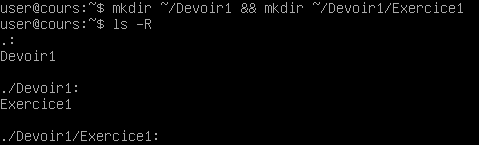
1. Pour créer un répertoire, il faut utiliser la commande mkdir. On veut en créer un dans notre répertoire d’accueil. Pour cela : soit on utilise la variable d’environnement $HOME, soit on utilise le tilde « ~ ». L’objectif est de venir créer, dans notre répertoire d’accueil, le répertoire Devoir1 et dans celui-ci, le répertoire Exercice1. Ce qui nous donnera les deux commandes suivantes :

mkdir ~/Devoir1

mkdir ~/Devoir1/Exercice1

Pour éviter de les exécuter une à une, on peut venir réunir les deux commandes sur une seule ligne avec « && » (permet d’exécuter la première commande et si le code de retour de celle-ci est égal à 0, alors la seconde commande est exécutée) ou « ; » (permet d’exécuter les commandes de manière séquentielle, la première est exécuter et la fin de celle-ci est attendu avant d’exécuter la seconde).

Ce qui nous donne : mkdir ~/Devoir1 && mkdir ~/Devoir1/Exercice1

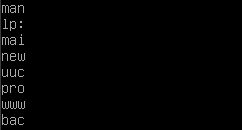


1. Pour afficher les 3 premiers caractères de chaque ligne du fichier /etc/passwd, il faut dans un premier temps utiliser la commande cat pour afficher le contenu de ce dernier. Puis on va utiliser un pipe « | » qui va permettre de rediriger la sortie d’une commande vers l’entrée d’une autre commande. La deuxième commande sera cut avec l’option -c qui permet de découper chaque ligne par caractères, en précisant l’intervalle 1-3.

Ce qui nous donne : cat /etc/passwd | cut –c1-3



Extrait du résultat :



1. Pour afficher les 10 premiers noms de logins du fichier /etc/passwd, il faut, comme précédemment, afficher le contenu du fichier avec la commande cat. On utilise une pipe afin de rediriger les lignes du fichier précédent vers la commande head –n, qui va nous permettre de récupérer les n premières lignes (ici, n sera égal à 10). On redirige, avec une pipe, ces 10 premières lignes vers la commande cut -f1 -d":", cela va nous permettre de récupérer le premier champ de la ligne délimité par un « : ».

Ce qui nous donne : cat /etc/passwd | head –n 10 | cut -f1 -d":"



1. Chaque ligne du fichier /etc/passwd représente un utilisateur, il suffit d’utiliser la commande wc -l pour compter le nombre de lignes du contenu du fichier affiché avec la commande cat, les deux commandes reliées pas une pipe. Dans l’objectif d’afficher le nombre d’utilisateurs.

Ce qui nous donne : cat /etc/passwd | wc -l



1. Pour obtenir la liste des utilisateur (nom prénom) triée par ordre alphabétique dans le fichier « utilisateur » du répertoire ~/Devoir1/Exercice1. Il faut dans un premier temps afficher le contenu du fichier /etc/passwd, puis récupérer le 5ème champ et trier par ordre alphabétique avec le résultat obtenu. Enfin, on redirige la sortie standard de la dernière commande vers ~/Devoir1/Exercice1/utilisateur.

Ce qui nous donne : cat /etc/passwd | cut -f5 -d":" | sort > ~/Devoir1/Exercice1/utilisateur

Présence du fichier créé :



Contenu du fichier créé (affiché avec la commande cat) :

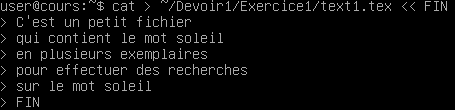


*On peut voir que certaines lignes sont vides, cela s’explique par le fait que rien n’a été saisi dans ce champ au moment de la création de l’utilisateur.*

Q2 :

1. Pour créer un fichier, il existe différentes méthodes utilisant cat, vi, nano ou encore touch. La méthode utilisé ici est celle avec cat. Pour cela, il suffit de rediriger la sortie standard cat vers le fichier que l’on souhaite créer. On vient ensuite ajouter la commande suivante sur la même ligne que cat : << FIN. Cela permet lorsqu’on saisit le mot préciser après << (ici FIN), d’arrêter la saisi. Ce mot ne sera pas présent dans le fichier.

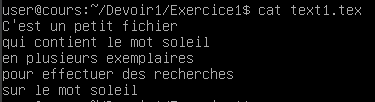
Ce qui nous donne : cat > ~/Devoir1/Exercice1/text1.tex << FIN



Présence du fichier créé :



Affichage de son contenu :



1. Pour copier le fichier « text1.tex » précédemment créé dans deux fichier « 1-text.tex » et « 2-text.tex », il faut utiliser la commande cp. Il faut en réaliser deux qui seront réuni sur la même ligne à l’aide de « ; ».

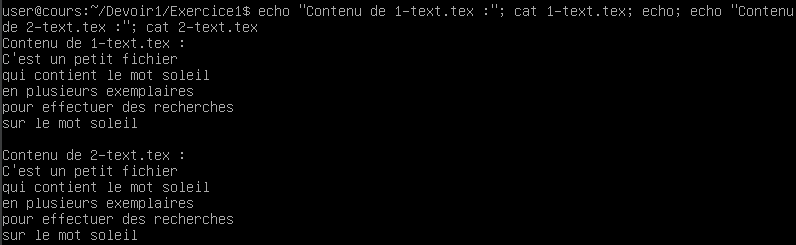
Ce qui donne : cp ~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/1-text.tex; cp ~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/2-text.tex



Présence des deux fichiers :



Contenu des deux fichiers :



1. Pour créer un fichier vide, on va utiliser la commande touch. Il suffit de lui préciser le chemin où se trouveront les futurs fichiers.

Ce qui donne : touch ~/Devoir1/test1.core; touch ~/Devoir1/Exercice1/test2.core



Présence du fichier « test1.core » dans le répertoire Devoir1 :



Contenu du fichier « test1.core » :



*On voit que le fichier est bien vide.*

Présence du fichier « test2.core » dans le répertoire Devoir1/Exercice1 :



Contenu du fichier « test2.core » :



*On voit que le fichier est bien vide.*

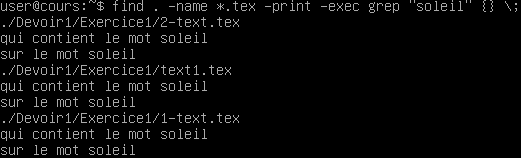
1. Pour rechercher une chaine dans un fichier, il faut utiliser la commande grep. Cette dernière permet, via l’utilisation d’une expression régulière (regular expression ou RegEx), de ressortir les lignes contenant les occurrences recherchées.

Ce qui nous donne : grep "soleil" ~/Devoir1/Exercice1/text1.tex



1. Pour rechercher tous les fichiers du répertoire courant (.) ayant dans leurs noms l’extension « .tex » (-name), il faut utiliser la commande find. Par la suite, il faut que cette dernière puisse exécuter (-exec) la commande grep afin de rechercher la chaine « soleil » dans tous les fichiers trouvés.

Ce qui nous donne : find . -name \*.tex -print –exec grep "soleil" {} \;



1. Pour recherche dans les fichiers du répertoire de l’utilisateur courant (~) les fichiers ayant dans leurs noms l’extension « .core » (-name), il faut utiliser la commande find.

Ce qui nous donne : find ~ -name \*.core -print



1. Pour supprimer tous les fichiers ayant l’extension « .core » dans le répertoire de l’utilisateur courant, il suffit de reprendre la commande précédente et de rajouter « -exec rm {} \; » en supprimant « -print ».

Ce qui nous donne : find ~ -name \*.core -exec rm {} \;



Preuve de la suppression du fichier /Devoir1/test1.core :



*On remarque que le fichier « test1.core » n’est plus présent dans le répertoire Devoir1, il a donc bien été supprimé.*

Preuve de la suppression du fichier /Devoir1/Exercice1/test2.core :

**

*On remarque que le fichier « test2.core » n’est plus présent dans le répertoire Devoir1/Exercice1, il a donc bien été supprimé.*

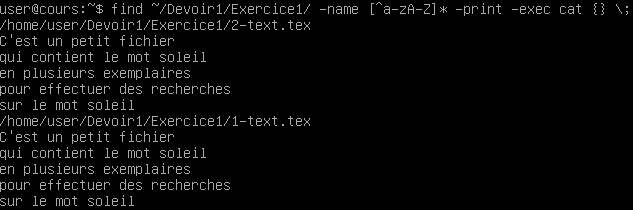
1. Pour afficher les fichiers du répertoire ~/Devoir1/Exercice1 dont le nom ne commence pas par une lettre, il faut utiliser la commande find ayant en option –name : [^a-zA-Z]\*.

Ce qui nous donne : find ~/Devoir1/Exercice1 -name [^a-zA-Z]\* -print



1. Pour afficher le contenu des fichiers du répertoire ~/Devoir1/Exercice1 dont le nom ne commence pas par une lettre, il faut reprendre la commande précédente et ajouter « -exec cat {} \; ».

Ce qui nous donne : find ~/Devoir1/Exercice1 -name [^a-zA-Z]\* -print -exec cat {} \;



Exercice 2)

Q1 : Tentative de suppression du fichier /etc/passwd avec la commande rm :



Affichage du fichier /etc/passwd et de ces informations avec la commande ls -l :



Affichage des informations sur le dossier /etc avec la commande ls -l :



Affichage des informations de l’utilisateur root présents dans /etc/passwd :



Affichage des informations du groupe principal de l’utilisateur root présents dans /etc/group :



On remarque que le fichier /etc/passwd ne s’est pas supprimé car la permission n’est pas accordée. En regardant les droits accordés sur ce fichier, on se rend compte que :

* L’utilisateur propriétaire a le droit de lecture (r) et d’écriture (w) sur ce fichier.
* Le groupe et les autres ont seulement un droit de lecture (r) sur ce fichier.

Et en regardant les droits accordées sur le répertoire /etc, on se rend compte que :

* L’utilisateur propriétaire a le droit de lire et d’afficher avec la commande ls le contenu du répertoire (r), de modifier le contenu (suppression et ajout de fichiers) du répertoire (w) et d’accéder aux fichiers contenus dans ce répertoire (x).
* Le groupe et les autres ont seulement le droit de lire et d’afficher le contenu du répertoire avec la commande ls (r) et d’accéder aux fichiers contenus dans ce répertoire (x).

Etant donné que nous somme l’utilisateur « user » (donc pas propriétaire du fichier /etc/passwd ni du répertoire /etc) et que nous n’appartenons pas au groupe principal de l’utilisateur propriétaire, les droits qui nous sont accordés sont ceux des autres (donc de lecture seulement sur le fichier et de lecture, d’affichage avec la commande ls ainsi que d’accès au contenu du répertoire /etc).

Les droits sur le fichier expliquent le premier message nous demandant si on souhaite supprimer un fichier protégé en écriture, car nous pouvons consulter le contenu du fichier mais nous ne pouvons pas le modifier.

Les droits sur le répertoire expliquent pourquoi nous ne pouvons pas supprimer le fichier, car nous n’avons ni le droit d’ajout ni le droit de suppression de fichiers (w) sur le répertoire /etc.

Q2 :

1. Pour qu’un fichier texte soit lisible mais non modifiable par tout le monde (y compris pour l’utilisateur propriétaire), il faut positionner les droits comme suivant : r--r--r--.

Pour se faire, on peut utiliser la commande chmod en écriture rwx de la manière suivant :

chmod a=r unFichier

Ou en écriture octal :

u : 100

g : 100

o : 100

100(2) = 22 \* 1 + 21 \* 0 + 20 \* 0 = 4 \* 1 + 0 + 0 = 4

Donc :

chmod 444 unFichier

1. Pour qu’un répertoire nommé « topsecret » n’ai son contenu modifiable et visible que par son utilisateur propriétaire, il faut positionner les droits comme suivant : rwx----.

Avec la commande chmod en écriture rwx :

chmod u=rwx g= o=g topsecret

En écriture octal :

u : 111

g : 000

o : 000

111(2) = 22 \* 1 + 21 \* 1 + 20 \* 1 = 4 \* 1 + 2 \* 1 + 1 \* 1 = 7

000(2) = 0

Donc :

chmod 700 topsecret

1. Pour qu’un utilisateur quelconque ne puisse pas lister les fichiers du répertoire « petitsecret » mais puisses les lires, il faut positionner les droits comme suivant : rwx--x--x.

Avec la commande chmod en écriture rwx :

chmod u=rwx g=x o=g petitsecret

En écriture octal :

u : 111

g : 001

o : 001

111(2) = 22 \* 1 + 21 \* 1 + 20 \* 1 = 4 \* 1 + 2 \* 1 + 1 \* 1 = 7

001(2) = 22 \* 0 + 21 \* 0 + 20 \* 1 = 0 + 0 + 1 \* 1 = 1

Donc :

chmod 711 petitsecret

Si un utilisateur quelconque appartient au groupe de l’utilisateur propriétaire, il aura les mêmes droits que les autres. Il pourra accéder au fichiers présent dans le répertoire petitsecret mais il ne pourra pas lister le contenu de ce dernier.

Q3 :

1. Pour que l’utilisateur *dupond* puisse recopier le fichier *f* situé dans son répertoire d’accueil, dans le répertoire d’accueil d’un utilisateur *antoine*. Il y a plusieurs conditions à remplir :

* Soit l’utilisateur *dupond*, appartient au groupe principal de l’utilisateur *antoine*. Dans c’est cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d’accueil (w) au groupe.
* Soit l’utilisateur *dupond* n’appartient pas au groupe principal de *antoine* mais il appartient à un groupe auquel ce dernier l’est aussi. Dans c’est cas-là, ce dernier modifie le groupe d’appartenance de son répertoire d’accueil avec chown et positionne le droit de modification du contenu de son répertoire d’accueil (w) à ce nouveau groupe.
* Soit *dupond* n’appartient à aucun groupe où *antoire* est présent. Dans c’est cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d’accueil (w) aux autres.

1. Pour que l’utilisateur *dupond* puisse effacer un fichier *fich1* situé dans le répertoire d’accueil d’un utilisateur *antoine*. Il y a plusieurs conditions à remplir :

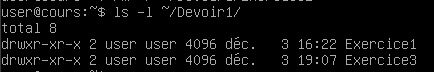
* Soit l’utilisateur *dupond*, appartient au groupe principal de l’utilisateur *antoine*. Dans c’est cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d’accueil (w) au groupe.
* Soit l’utilisateur *dupond* n’appartient pas au groupe principal de *antoine* mais il appartient à un groupe auquel ce dernier l’est aussi. Dans c’est cas-là, ce dernier modifie le groupe d’appartenance de son répertoire d’accueil avec chown et positionne le droit de modification du contenu de son répertoire d’accueil (w) à ce nouveau groupe.
* Soit *dupond* n’appartient à aucun groupe où *antoire* est présent. Dans c’est cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d’accueil (w) aux autres.

Exercice 3)

1. Création du sous-répertoire « Exercice3 » du répertoire ~/Devoir1 avec la commande mkdir :



Présence du répertoire « Exercice3 » dans le répertoire « Devoir1 » :



1. La commande permettant d’afficher « Bonjour » à l’écran et en même temps de créer un fichier « salutation.txt » dans le répertoire « Exercice3 », précédemment créé, dont le contenu est le même message, est la suivante :



Preuve de l’affichage du message « Bonjour » à l’écran :



Preuve de la création du fichier « salutation.txt » dans le répertoire ~/Devoir1/Exercice3 :



Et contenant le même message que celui affiché :

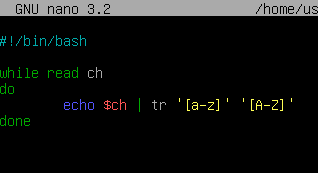


1. Création du script shell « lectEcriture » qui permet de passer les caractères en minuscule, en majuscule :

* Création du fichier du script avec nano :



* Code du script :



*La commande read permet de lire sur l’entrée standard et d’affecter la valeur à une variable, ici nommé ch. La boucle while permet ici de lire, si sur l’entrée standard il y a plusieurs lignes, chaque ligne (délimité par un retour à la ligne). On appel ensuite la commande echo avec cette dernière qui permet, en temps normal, d’affiché la valeur de la variable sur la sortie standard. Sauf qu’une pipe redirige la sortie standard de la commande echo vers l’entré standard de la commande tr. Cette dernière permet de remplacer chaque caractère dans l’intervalle [a-z] (les caractères en minuscules) en son équivalant dans l’intervalle [A-Z] (en majuscules).*

* Ajout du droit d’exécution sur le shell script « lectEcriture » pour l’utilisateur propriétaire (nous) :



* Ajout du chemin où se trouve « lectEcriture » dans la variable d’environnement PATH de manière à pouvoir exécuter le shell script en l’appelant uniquement par le nom du fichier :

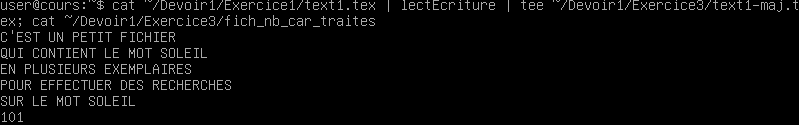


1. Modification du script précédent, afin qu’il écrive le nombre de caractères traités dans un fichier :

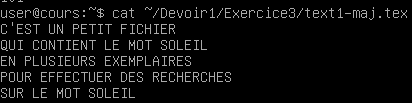


*La variable s\_ch contient le nombre de caractères réel de la chaine (-1 car le caractère de retour à la ligne est compté avec la commande wc -c). La variable i est un compteur qui va nous permettre de passer en revu chaque caractère de la chaine, elle est incrémentée à chaque itération dans la boucle while. Ces deux variables vont former la condition permettant de sortir de cette boucle (tant que la valeur de i est inférieure ou égale à celle de la taille, on reste dans la boucle). A chaque itération, on va venir regarder si chaque caractère de ch correspond à une minuscule (grep –Ec nous permet de compter le nombre de ligne ayant un ou plusieurs caractères correspondant à la RegEx. Ici, vu que l’on test cette dernière pour un seul caractère : si cette commande retourne 1, alors le caractère est une minuscule, 0 sinon). Si le caractère tester est une minuscule, on incrémente alors la variable cpt\_car\_traites. Cette dernière correspond au nombre de caractères traités. On finit par afficher la valeur de ch en majuscule sur la sortie standard et on utilise la commande echo en redirigeant sa sortie standard dans un fichier pour y écrire la valeur de cpt\_car\_traites.*

1. Pour recopier le contenu passé en majuscule du fichier ~/Devoir1/Exercice1/text1.tex dans un fichier text1-maj.tex (qui sera sauvegardé dans le répertoire ~/Devoir1/Exercice3) et l’afficher sur l’écran tout en affichant le nombre de caractères traités, il faut utiliser la commande suivante : cat ~/Devoir1/Exercice1/text1.tex | lectEcriture | tee ~/Devoir1/Exercice3/text1-maj.tex; cat ~/Devoir1/Exercice3/fich\_nb\_car\_traites



Contenu du fichier ~/Devoir1/Exercice3/text1-maj.tex :



Exercice 4)

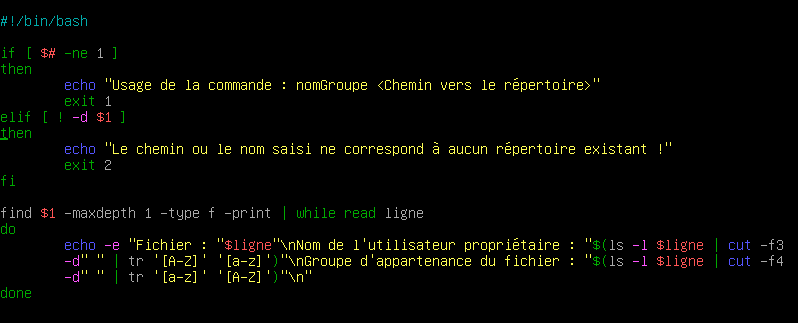
Q1 : Création du sous-répertoire « Exercice4 » dans le répertoire « Devoir1 » avec la commande mkdir :



Q2 : Création du script shell « nomGroupe » dans le répertoire ~/Devoir1/Exercice4 avec la commande nano :



Code du script shell « nomGroupe » :



Ajout du droit d’exécution sur le script shell précédent pour l’utilisateur propriétaire (nous) :

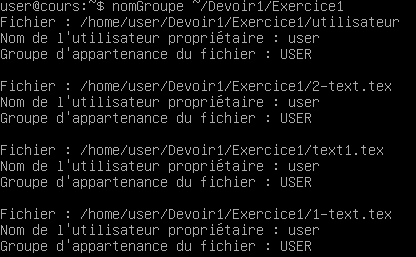


Ajout du chemin où se trouve le script shell précédent dans la variable d’environnement PATH, afin de pouvoir l’exécuter en utilisant le nom du fichier uniquement :



**Résultats :**

* Avec un chemin correct :



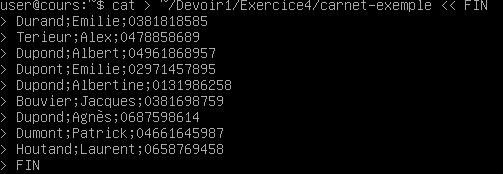
* Avec un chemin incorrect :



* Sans paramètre :



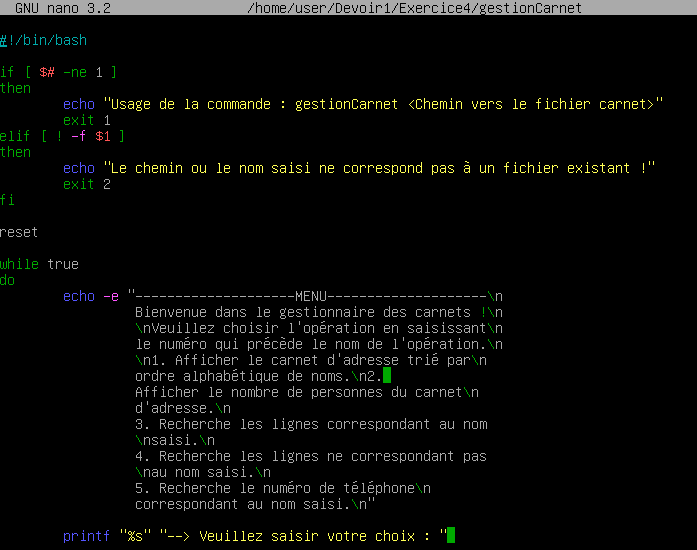
Q3 : Création d’un fichier dans le répertoire ~/Devoir1/Exercice4, contenant un exemple de type carnet d’adresse avec la commande cat qui nous permettra de tester le futur script gestionCarnet, nommé « carnet-exemple » :

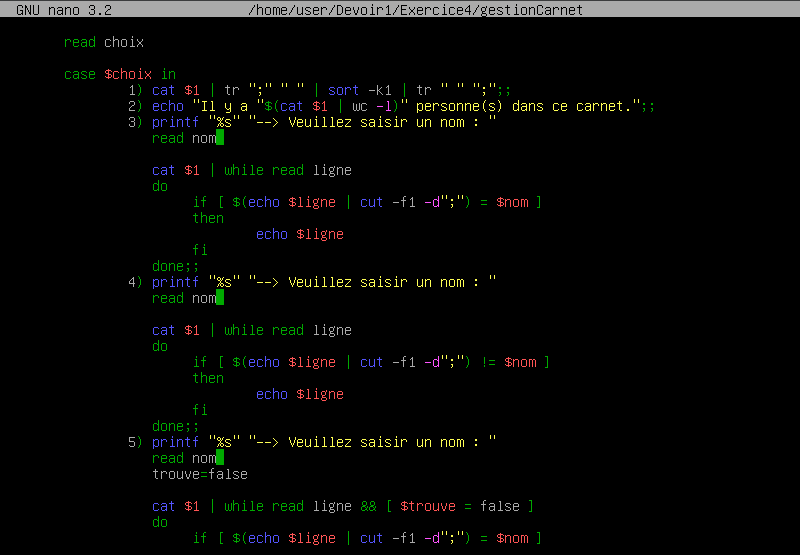


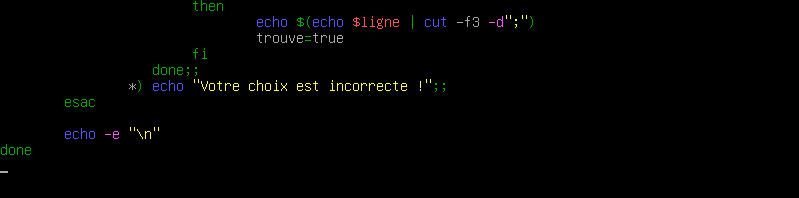
Création du script shell « gestionCarnet » dans le répertoire ~/Devoir1/Exercice4 avec nano :



Code du script précédemment créer :







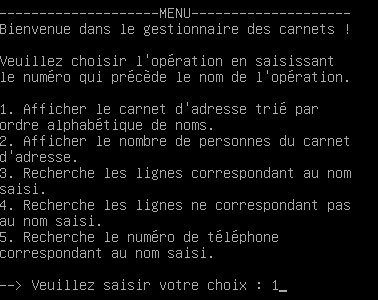
Ajout du droit d’exécution sur le script shell précédent pour l’utilisateur propriétaire (nous) :

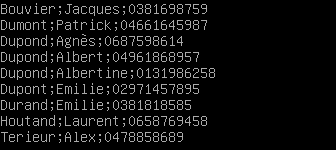


Pas besoins d’ajouter le chemin où se trouve le script shell dans la variable d’environnement PATH, car cela a déjà été fait à la question précédente.

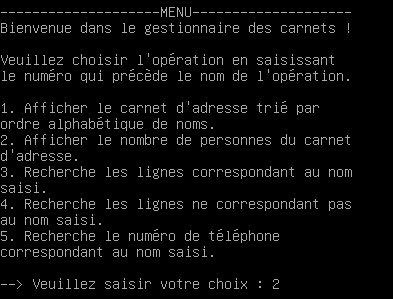
**Résultats :**

* Test du premier choix :



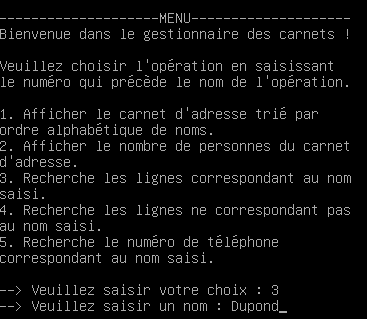


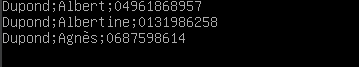
* Test du second choix :



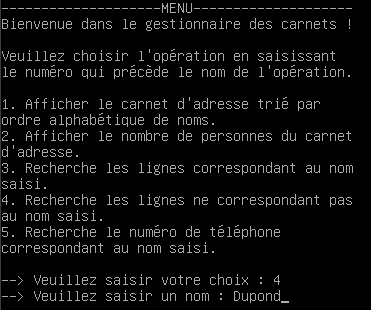


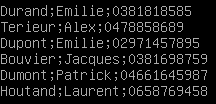
* Test du troisième choix :



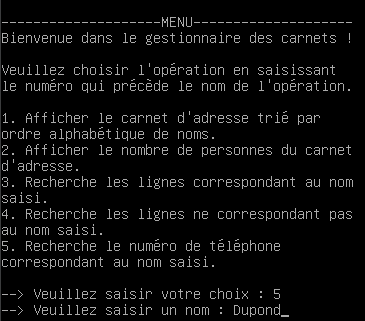


* Test du quatrième choix :





* Test du cinquième choix :





* Test d’un choix erroné :

