

Bloc 1 TP-21

Contexte	2
Question 1	2
Question 2	2
Question 3	3
Question 4	3
Question 5	3
Question 6	3
Question 7	4
Question 8	4
Question 9	4
Question 10	5
Question 11	5
Question 12	6
Question 13	6
Question 14	6
Question 15	7
Question 16	7
Question 17	8
Question 18	8
Question 19	10
Question 20	10
Conclusion	10

Contexte :

Dans le cadre du développement d'une nouvelle formule de gel nettoyant par l'entreprise **Canard WC**, vous êtes chargé, en tant qu'administrateur système, de mettre en place l'environnement de travail sécurisé pour l'équipe de recherche.

Procédure technique d'intervention :

1. Tout d'abord, l'administrateur doit passer en super utilisateur pour avoir tous les droits sur la machine.



```
aubree@ubuntu:~/cp$ sudo su
[sudo] password for aubree:
root@ubuntu:/home/aubree/cp# _
```

2. Il doit commencer par accueillir les deux nouveaux employés en créant les utilisateurs Donald et Daisy.



```
Ajout de l'utilisateur « donald » ...
Ajout du nouveau groupe « donald » (1001) ...
Ajout du nouvel utilisateur « donald » (1001) avec le groupe « donald » ...
Création du répertoire personnel « /home/donald »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
Changing the user information for donald
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Ces informations sont-elles correctes ? [0/n] o
root@ubuntu:/home/aubree/cp# adduser daisy
Ajout de l'utilisateur « daisy » ...
Ajout du nouveau groupe « daisy » (1002) ...
Ajout du nouvel utilisateur « daisy » (1002) avec le groupe « daisy » ...
Création du répertoire personnel « /home/daisy »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
Changing the user information for daisy
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Ces informations sont-elles correctes ? [0/n]
root@ubuntu:/home/aubree/cp# _
```

3. Pour structurer l'équipe de recherche, il doit créer un groupe de travail nommé "laboratoire".

```
7 directories, 24 files
root@ubuntu:~# addgroup laboratoire
Ajout du groupe « laboratoire » (GID 1003)...
Fait.
root@ubuntu:~#
```

4. Le chimiste Donald doit pouvoir travailler avec l'équipe. Il faut donc ajouter l'utilisateur Donald au groupe laboratoire.

```
root@ubuntu:~# usermod -aG laboratoire donald
root@ubuntu:~#
```

5. L'administrateur doit maintenant préparer l'espace de travail. Il se rend dans le dossier /home et crée le répertoire du projet nommé "Projet_Canard_X".

```
root@ubuntu:~# cd /home
root@ubuntu:/home# mkdir Project_Canard_X
root@ubuntu:/home# ls
aubree daisy donald Project_Canard_X
root@ubuntu:/home#
```

6. À l'intérieur du projet, il doit organiser le travail en créant trois sous-dossiers : "Formules", "Tests_Echoues" et "Archives".

```
root@ubuntu:/home# cd Project_Canard_X/
root@ubuntu:/home/Project_Canard_X# mkdir Formules Tests_Echoues Archives
```

7. Il souhaite vérifier visuellement que toute l'arborescence des dossiers a été correctement créée.

```
root@ubuntu:/home/Project_Canard_X# tree
.
├── Archives
├── Formules
└── Tests_Echoues

3 directories, 0 files
root@ubuntu:/home/Project_Canard_X#
```

8. Il crée ensuite un premier fichier vide pour la recette dans le dossier Formules, nommé "recette_bleue.txt".

```
root@ubuntu:/home/Project_Canard_X# touch Formules/recette_bleue.txt
root@ubuntu:/home/Project_Canard_X# ls Formules/
recette_bleue.txt
```

9. Par mesure de sécurité, il doit faire une copie de ce fichier dans le dossier Archives sous le nom "recette_bleue.bak".

```
root@ubuntu:/home/Project_Canard_X# cp Formules/recette_bleue.txt Archives/recette_bleue.bak
root@ubuntu:/home/Project_Canard_X# tree
.
├── Archives
│   └── recette_bleue.bak
├── Formules
│   └── recette_bleue.txt
└── Tests_Echoues

3 directories, 2 files
root@ubuntu:/home/Project_Canard_X#
```

10. On décide de changer le nom de cette sauvegarde. Il doit renommer le fichier dans le dossier Archives pour qu'il s'appelle "recette_finale.txt".

```
root@ubuntu:/home/Project_Canard_X# mv Archives/recette_bleue.bak Archives/recette_finale.txt
root@ubuntu:/home/Project_Canard_X# tree
.
├── Archives
│   └── recette_finale.txt
├── Formules
│   └── recette_bleue.txt
└── Tests_Echoues

3 directories, 2 files
root@ubuntu:/home/Project_Canard_X# _
```

11. Finalement, on veut utiliser cette version. Il déplace le fichier "recette_finale.txt" du dossier Archives vers le dossier Formules.

```
root@ubuntu:/home/Project_Canard_X# mv Archives/recette_finale.txt Formules/
root@ubuntu:/home/Project_Canard_X# tree
.
├── Archives
├── Formules
│   ├── recette_bleue.txt
│   └── recette_finale.txt
└── Tests_Echoues

3 directories, 2 files
root@ubuntu:/home/Project_Canard_X#
```

12. Pour gérer la sécurité, il attribue la propriété du dossier complet du projet à l'utilisateur Donald et au groupe laboratoire.

```
root@ubuntu:/home/Project_Canard_X# cd /home
root@ubuntu:/home# chown -R donald:laboratoire Project_Canard_X
root@ubuntu:/home# ls Project_Canard_X/ -la
total 20
drwxr-xr-x 5 donald laboratoire 4096 nov. 25 15:45 .
drwxr-xr-x 6 root root 4096 nov. 25 15:43 ..
drwxr-xr-x 2 donald laboratoire 4096 nov. 25 15:54 Archives
drwxr-xr-x 2 donald laboratoire 4096 nov. 25 15:54 Formules
drwxr-xr-x 2 donald laboratoire 4096 nov. 25 15:45 Tests_Echoues
root@ubuntu:/home# _
```

13. Il doit maintenant modifier les permissions pour que le propriétaire ait tous les droits, le groupe puisse seulement lire et entrer dans les dossiers, et les autres n'aient aucun accès.

```
root@ubuntu:/home# chmod -R 750 Project_Canard_X
root@ubuntu:/home# ls Project_Canard_X/ -la
total 20
drwxr-x--- 5 donald laboratoire 4096 nov. 25 15:45 .
drwxr-xr-x 6 root root 4096 nov. 25 15:43 ..
drwxr-x--- 2 donald laboratoire 4096 nov. 25 15:54 Archives
drwxr-x--- 2 donald laboratoire 4096 nov. 25 15:54 Formules
drwxr-x--- 2 donald laboratoire 4096 nov. 25 15:45 Tests_Echoues
root@ubuntu:/home# _
```

14. L'administrateur va vérifier les accès en se connectant sur la session de Donald et en entrant dans le dossier du projet.

```
root@ubuntu:/home# su donald
donald@ubuntu:/home$ cd /home/Project_Canard_X/
donald@ubuntu:/home/Project_Canard_X$
```

15. En tant que Donald, il simule une expérience ratée en créant un fichier de log nommé “test_acide.log” dans le dossier Tests_Echoues.

```
donald@ubuntu:/home/Project_Canard_X$ touch Tests_Echoues/test_acide.log
donald@ubuntu:/home/Project_Canard_X$ tree
.
├── Archives
├── Formules
│   ├── recette_bleue.txt
│   └── recette_finale.txt
└── Tests_Echoues
    └── test_acide.log

3 directories, 3 files
donald@ubuntu:/home/Project_Canard_X$
```

16. Le test est fini. Il nettoie ses traces en supprimant le fichier “test_acide.log”, puis il supprime le dossier Tests_Echoues devenu vide.

```
donald@ubuntu:/home/Project_Canard_X$ rm Tests_Echoues/test_acide.log
donald@ubuntu:/home/Project_Canard_X$ rmdir Tests_Echoues/
donald@ubuntu:/home/Project_Canard_X$ tree
.
├── Archives
├── Formules
│   ├── recette_bleue.txt
│   └── recette_finale.txt
└──

2 directories, 2 files
donald@ubuntu:/home/Project_Canard_X$ _
```

17. Il se déconnecte de la session de Donald pour repasser administrateur, puis il sauvegarde tout le travail en compressant le dossier du projet dans une archive nommée "livraison_canard.tar.gz".

```
donald@ubuntu:/root$ exit
exit
root@ubuntu:~# cd /home
root@ubuntu:/home# tar -czvf livraison_canard.tar.gz Project_Canard_X
Project_Canard_X/
Project_Canard_X/Formules/
Project_Canard_X/Formules/recette_bleue.txt
Project_Canard_X/Formules/recette_finale.txt
Project_Canard_X/Archives/
root@ubuntu:/home#
```

18. Pour tester la stabilité du système, il lance un programme Python en boucle infinie (dans un terminal), et repère son identifiant (PID) en utilisant la commande de surveillance (dans un autre terminal).

```
root@ubuntu:~# python3
Python 3.4.3 (default, Nov 12 2018, 22:25:49)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> while True:
...     print("INVASION DES MOUCHES MALICIEUSES")
... _
```



```

INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES
INVASION DES MOUCHES MALICIEUSES

```



```

top - 15:07:01 up 12 min, 3 users, load average: 1,90, 1,42, 0,74
Tâches: 169 total, 2 en cours, 167 en veille, 0 arrêté, 0 zombie
%Cpu(s): 7,1 ut, 31,1 sy, 0,0 ni, 61,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 4028204 total, 247088 used, 3781116 free, 26212 buffers
KiB Swap: 4194300 total, 0 used, 4194300 free. 102300 cached Mem

```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
1404	root	20	0	34056	8308	5348	R	91,0	0,2	1:36.88	python3
1	root	20	0	33396	3932	2692	S	0,0	0,1	0:02.09	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	ksoftirqd/0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0,0	0,0	0:00.23	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/1
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/1
13	root	20	0	0	0	0	S	0,0	0,0	0:00.03	ksoftirqd/1
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0H
16	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
17	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	netns
18	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	perf
19	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khungtaskd
20	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	writeback
21	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
22	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
23	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	crypto
24	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kinTEGRITYd
25	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	bioSet
26	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kblockd
27	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	ata_sff
28	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	md
29	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	devfreq_wq
31	root	20	0	0	0	0	S	0,0	0,0	0:00.07	kworker/0:1
33	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kswapd0
34	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	vmstat

19. La simulation consomme trop de ressources. Il force l'arrêt immédiat de ce processus bloqué en utilisant son PID.



```
root@ubuntu:/home/aubree# kill -9 1404
root@ubuntu:/home/aubree# _
```

20. Enfin, il termine son intervention en affichant la liste de toutes ses actions passées pour prouver son travail.



```
donald@ubuntu:/home/aubree$ history
1 cd /home/Project_Canard_X/
2 touch Tests_Echoues/test_acide.log
3 tree
4 rm Tests_Echoues/test_acide.log
5 rmdir Tests_Echoues/
6 tree
7 exit
8 cd
9 exit
10 cd
11 exit
12 cd /home/Project_Canard_X/
13 exit
14 history
donald@ubuntu:/home/aubree$
```

Conclusion

Ce TP nous a placés en situation professionnelle pour concevoir une situation et un contexte complets pour une entreprise fictive. L'élaboration de ce scénario a permis de valider l'utilisation de commandes Linux au sein d'une chaîne d'intervention logique.