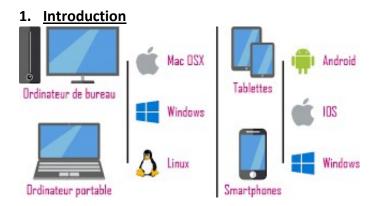
PARTIE 3: Les systèmes d'exploitation





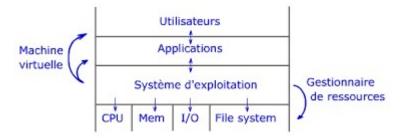
DÉFINITION 1:

Le système d'exploitation est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur (le hardware) et les applications de l'utilisateur (le software).

Ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations spécifiques au périphérique, il lui suffit d'envoyer les informations au système d'exploitation, qui se charge de les transmettre au périphérique concerné via son pilote. En l'absence de pilotes il faudrait que chaque programme reconnaisse et prenne en compte la communication avec chaque type de périphérique!

Pour ce qui est du développement des applications : l'OS propose une interface de programmation appelée API

(Application Program Interface), une sorte de boîte à outils à laquelle les développeurs recourent pour construire leurs applications. Ces API procurent une vue uniforme et simplifiée des ressources de la machine. Cela permet aux applications de faire abstraction des particularités du matériel en dissimulant la diversité et la complexité du hardware.



Il existe divers systèmes d'exploitation qui peuvent se découper en deux familles de systèmes :

- Les systèmes propriétaires : Windows, MacOS, UNIX, ...
- Les systèmes libres : Linux, Android, ...



DÉFINITION 2:

La différence essentielle est que le code d'un logiciel libre (et donc d'un système libre) est public. On peut en général le modifier ou s'en servir pour fabriquer de nouveaux produits. Les logiciels propriétaires sont en général non ouverts, il est donc plus difficile (voire illégal) de les modifier.

Attention qui dit logiciel libre ne veut pas forcement dire logiciel gratuit (même si c'est souvent le cas), la confusion entre "libre" et "gratuit" vient de l'anglais : "free" veut à la fois dire "libre", mais aussi gratuit.

Vous connaissez déjà quelques systèmes d'exploitation :



Un peu d'histoire des systèmes d'exploitation :

1. A la préhistoire des systèmes d'exploitation, il n'y avait pas d'interface graphique. MS-DOS est un des premiers systèmes d'exploitation grand public.

```
HIMEM: DOS XMS Driver, Version 2.78 - 09/19/91
XMS Specification Version 2.0
Copyright 1988-1991 Microsoft Corp.
Installed A20 handler number 2.
64K High Memory Area is available.
MODE prepare code page function completed
MODE select code page function completed
C:\>uer
MS-DOS Version 5.00
C:\>_
```

On retrouve encore un soupçon de MS-DOS dans l'invite de commande Windows

2. Microsoft a été créée par Bill Gates et Paul Allen en 1975. Microsoft est surtout connue pour son système d'exploitation Windows. Windows est un système d'exploitation "propriétaire". La première version de Windows date de 1983 et n'est qu'un ajout sur le système MS-DOS.



- 3. Les systèmes d'exploitation Apple. Tous les ordinateurs d'Apple sont livrés avec le système d'exploitation MacOs. Ce système MacOs est un système d'exploitation UNIX, c'est donc un système d'exploitation propriétaire.
- 4. Le système UNIX est un système propriétaire (son code source n'est pas disponible, l'étude d'UNIX est donc impossible).

En 1991, un étudiant finlandais, Linus Torvalds, décide de créer un clone libre d'UNIX en ne partant de rien (on dit "from scratch" en anglais) puisque le code source d'UNIX n'est pas public. Ce clone d'UNIX va s'appeler Linux (Linus+UNIX).



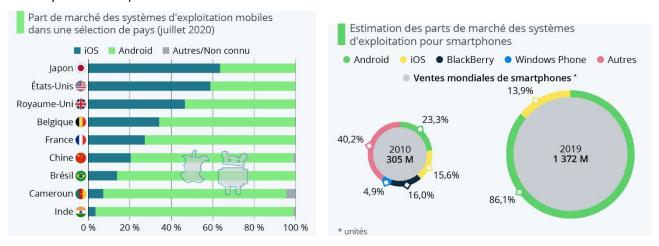
En conclusion des systèmes d'exploitation pour ordinateurs, voici les parts de marché des systèmes d'exploitation au cours du mois de mai 2020.

Part de marché mondiale des systèmes d'exploitation PC

(Mai 2020, source NetMarketShare)



5. Les systèmes d'exploitation mobiles.



Pour compléter cette introduction, vous pouvez visualiser les deux vidéos suivantes :

- Histoire des systèmes d'exploitation (netprof) : https://www.youtube.com/watch?v=4OhUDAtmAUo
- Histoire du système d'exploitation Unix (netprof) : https://www.youtube.com/watch?v=Za6vGTLp-wg
 Dans le cadre de la NSI nous utiliserons des distributions Windows et GNU/Linux.

PARTIE 4 : A la découverte d'un nano ordinateur : Raspberry Pi

1. Le Raspberry Pi

1.1. Présentation

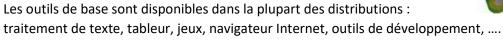
Le Raspberry Pi est un nano ordinateur, c'est-à-dire un ordinateur de la taille d'une carte de crédit, coûtant environ 40 € et capable de diffuser des vidéos en 1080p (Full HD 1920x1080

pixels) à 30 images par seconde.

Nous réaliserons nos activités pratiques avec le Raspberry Pi 3 ou bien sur une machine virtuelle.

Le Raspberry Pi 3 regroupe sur une carte de 85 ×56 mm tous les composants nécessaires pour faire tourner un système d'exploitation et utiliser ainsi le nano ordinateur comme un véritable PC.

Les outils de base sont disponibles dans la plupart des distributions :



1.2. Historique

Au début des années 2000, un petit groupe de personnes a l'ambition de rendre accessible au plus grand nombre de jeunes la programmation informatique. Pour satisfaire cette ambition, il fallait trouver une plateforme qui, à l'instar des anciens ordinateurs, pourrait démarrer directement dans un environnement de programmation. Pour contrer le fait qu'on ne peut pas laisser un jeune expérimenter ses programmes sur un PC, il faut une machine d'un prix tel que la notion de "risque" disparaisse.

À partir de 2008, les microprocesseurs conçus pour des utilisations mobiles sont devenus plus abordables, mais aussi suffisamment puissants pour fournir une vidéo d'excellente qualité. Le groupe a donc pensé que cette caractéristique pourrait rendre la carte attirante pour les jeunes, pas forcément intéressés par un outil destiné à la programmation pure.

Le projet semblait devenir réalisable. Eben Upton (aujourd'hui concepteur de composant chez Broadcom), Robert Mullins, Jack Lang et Alan Mycroft en équipe avec Pete Lomas, directeur de l'ingénierie chez Norcott Technologies, ainsi que David Braben, auteur du jeu Elite pour le BBC Micro, ont créé la Fondation Raspberry Pi pour que le projet devienne une réalité.

Trois ans plus tard, la fabrication en série commence, grâce à des accords de fabrication sous licence signés avec Farnell element14 et RS Components. Et ce n'est que le début de l'histoire du Raspberry Pi.

La Fondation Raspberry Pi, équivalent d'une association loi de 1901 en France, explique la genèse du projet Raspberry Pi sur son site web (<u>www.raspberrypi.org/about</u>).

1.3. Le logo

En dehors des considérations techniques et financières, la Fondation se préoccupait également de trouver une image pour la représenter. Il ne faisait aucun doute que le succès du Raspberry Pi lors de sa sortie allait avoir un retentissement dans les médias. Il fallait pouvoir identifier visuellement le produit.

Démarrée le 5 août 2011, la compétition posait comme contrainte que le logo devait être reconnaissable dans un carré de 1 cm par 1 cm, mais que sa qualité devait également permettre de l'utiliser sur différents produits de tailles différentes. Enfin, il devait être identifiable facilement aussi bien en couleur qu'en noir et blanc. Le 7 octobre de la même année, Liz Upton publiait le résultat sur le site de la Fondation. Le logo dessiné par Paul Beech avait été retenu après le vote du jury.

Le choix de ce logo n'est pas innocent : il représente bien une framboise, mais sous la forme d'un fullerène C60, composé d'atomes de carbone. Ce fullerène de forme sphérique (ce qui rappelle le nombre Pi) est composé de 20 hexagones et 12 pentagones, soit 32 faces. Or le processeur du Raspberry Pi est un processeur 32 bits. De plus, sur ces 32 faces, 11 sont visibles sur le logo. Et le processeur du Raspberry Pi est un ARM... 11! Difficile de trouver plus de coïncidences.

Nous étudierons plus loin dans cette séquence, l'architecture matérielle du Raspberry Pi et de son processeur.

2. A la découverte de Raspbian

2.1. Qu'est-ce que c'est?

Pour faire fonctionner un ordinateur, il faut un système d'exploitation. Cet OS (Operating System ou système d'exploitation) est le chef d'orchestre qui gère les ressources de l'ordinateur et les attribue aux diverses tâches en cours d'exécution, que ce soient des applications lancées par l'utilisateur ou des services propres au système luimême.

Lorsqu'un logiciel a besoin d'une ressource matérielle, c'est le système d'exploitation qui la lui attribue. L'OS gère le matériel (mémoire et périphériques), l'accès aux fichiers stockés sur les mémoires de masse, ainsi que l'échange d'informations sur le réseau.

Lorsque la Fondation a créé le Raspberry Pi, le maintien du prix sous la barre fixée au départ impliquait qu'il n'y ait pas de licence à payer pour l'utilisation du système d'exploitation. C'est donc tout naturellement que les créateurs se sont tournés vers Linux.

2.2. Linux

Linux est un système d'exploitation libre sous licence GPL (General Public License = licence publique générale). C'est-à-dire que les codes sources des programmes sont à la disposition de tout un chacun. N'importe qui peut utiliser Linux, reprendre les codes, les étudier, les modifier et les redistribuer. Ce sont des milliers de développeurs, parfois rémunérés par des entreprises, qui participent à l'évolution de Linux.



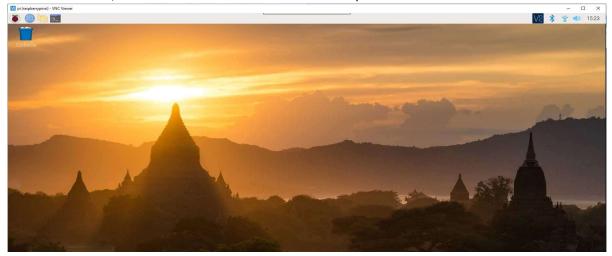
2.3. <u>Découvrir l'OS Raspbian</u>

Nous utiliserons une machine virtuelle* avec VMWARE Player (démonstration par le professeur).

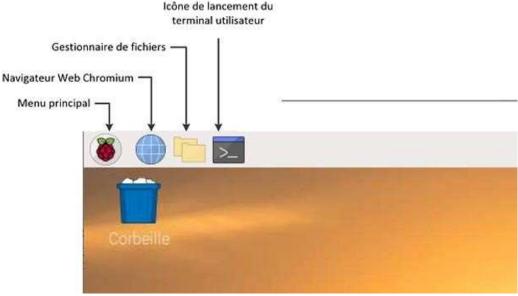
*Une machine virtuelle (anglais virtual machine, abr. VM) est une illusion d'un appareil informatique créée par un logiciel d'émulation. Le logiciel d'émulation simule la présence de ressources matérielles et logicielles telles que la mémoire, le processeur, le disque dur, voire le système d'exploitation et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions que celles de la machine simulée.

2.3.1. Utilisation en mode graphique

Une fois connecté, vous devez obtenir l'écran d'accueil Raspbian suivant :



La barre des tâches comporte une partie gauche (ci-dessous), une zone centrale affichant les fenêtres ouvertes sur le bureau ou mises en réduction et une partie droite donnant des informations sur les connexions réseau actives ou encore l'heure.



Prenez un peu de temps pour découvrir les menus offerts par la barre de tâches.

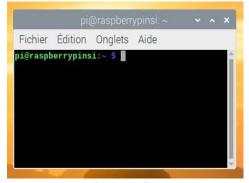
2.3.2. Utilisation de la ligne de commande

À la "préhistoire" des systèmes d'exploitation, ces derniers étaient dépourvus d'interface graphique (système de fenêtres "pilotables" à la souris), toutes les interactions "système d'exploitation - utilisateur" se faisaient par l'intermédiaire de "lignes de commandes" (suites de caractères, souvent ésotériques, saisies par l'utilisateur). Aujourd'hui, même si les interfaces graphiques modernes permettent d'effectuer la plupart des opérations, il est important de connaître quelques-unes de ces lignes de commandes.

L'utilisation du mode « texte » est certes moins convivial mais il permet une meilleure compréhension du fonctionnement du système d'exploitation Raspbian. C'est par conséquent le mode que nous allons privilégier dans cette activité.

EXERCICE 1:

a. Ouvrez une fenêtre « Terminal de commande » en cliquant sur l'icône « LXTermainal » :



b. Le système de fichiers de Raspbian (Linux) est organisé à partir d'un point de départ appelé root, racine ou encore /.

Sous cette racine se déploient des répertoires contenant les fichiers et programmes nécessaires au système d'exploitation.

- c. À tout moment, situez-vous dans cette arborescence en tapant la commande « pwd »
- d. Remontez au plus haut niveau de l'arborescence en tapant la commande « cd / » (cd pour « change working directory »)
- e. Découvrez le nom des répertoires (« directories ») situés à la racine en tapant la commande « ls »
- f. Combien de répertoires figurent à la racine de l'arborescence ?
- g. Visualisez l'arborescence de votre système avec la commande « tree -L 1 ». Essayer en remplaçant "1" par "2". Qu'observe-t-on?

```
bin
boot
dev
etc
home
initrd.img -> boot/initrd.img-4.19.0-13-amd64
initrd.img.old -> boot/initrd.img-4.19.0-13-amd64
lib
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
Lmp
usr
var
vmlinuz -> boot/vmlinuz-4.19.0-13-amd64
vmlinuz.old -> boot/vmlinuz-4.19.0-13-amd64
```

- h. Atteignez le répertoire /home avec la commande « cd home ».
- i. Entrer à nouveau la commande « ls ». Quel résultat obtenez-vous ?



PROPRIÉTÉ 1 : Le contenu des principaux répertoires de l'arborescence « Linux »

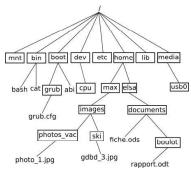
1	Racine ou root, contient les répertoires de l'arborescence Linux.
bin	Exécutables binaires du système cp, ls, mount, rm
boot	Fichiers de démarrage de Linux.
dev	Fichiers spéciaux assurant la liaison avec les périphériques.
etc	Fichiers de configuration du système, des services
home	Répertoire personnel des utilisateurs.
lib	Bibliothèques système partagées.
media	Point de montage des clés USB, CD-ROM
mnt	Point de montage temporaire de partitions et périphériques.
proc	Informations sur les processus et le noyau Linux.
root	Répertoire personnel du super-utilisateur.
sbin	Binaires système et outils comme fsck.
tmp	Fichiers temporaires.
usr	Fichiers binaires et commandes utilisateurs.
var	Système de fichiers "variables" (modifiables) ; on y trouve le contenu web (répertoire www), mais aussi les logs (journaux).

Ces répertoires contiennent eux-mêmes d'autres répertoires qui constituent l'arborescence de Linux. Quelques différences peuvent exister entre les distributions, mais la plus grande partie de l'arborescence est commune.

- .. répertoire parent
- ~ répertoire « home » de l'utilisateur

/ racine du système

3. Se déplacer dans l'arborescence



Pour indiquer la position d'un fichier (ou d'un répertoire) dans l'arborescence, il existe 2 méthodes : indiquer un chemin absolu ou indiquer un chemin relatif.

3.1. Chemin absolu

Le chemin absolu doit indiquer "le chemin" depuis la racine. Par exemple le chemin absolu du fichier fiche.ods sera : /home/elsa/documents/fiche.ods

A EXERCICE 2:

- Donner le chemin absolu pour accéder au fichier gdbd_3.jpg
- Donner le chemin absolu pour accéder au fichier rapport.odt
- Donner la commande à taper pour accéder au répertoire photos_vac

3.2. Chemin relatif

Il est possible d'indiquer le chemin non pas depuis la racine, mais depuis un répertoire quelconque, nous parlerons alors de chemin relatif.

Le chemin relatif permettant d'accéder au fichier photo_1.jpg depuis le répertoire "max" est : images/photo_vac/photo_1.jpg

Le chemin relatif permettant d'accéder au fichier photo_1.jpg depuis le répertoire "document" est :

../../max/images/photo_vac/photo_1.jpg

EXERCICE 3:

- Donner le chemin relatif pour accéder au fichier gdbd_3.jpg depuis "home"
- Donner le chemin relatif pour accéder au fichier rapport.odt depuis "max"
- Donner la commande à taper pour accéder au répertoire photos_vac depuis "documents"

A EXERCICE 4:

- Positionnez-vous dans le répertoire \etc
- Taper la commande « ls -al » . Que permet-elle de visualiser ?
- Comment reconnait-on les répertoires des fichiers autrement que par la couleur de l'écriture ?
- Taper la commande « ls -al | more » afin de pouvoir visualiser le contenu du répertoire. Enfoncez la barre d'espace pour activer le défilement.
- D'après-vous, python est-il un fichier ou un répertoire? Comment le vérifier?
- Déplacez-vous dans le répertoire "xml".
- Quelle commande doit-on taper pour se déplacer vers /etc/vim En chemin relatif :

En chemin absolu:



PROPRIÉTÉ 2 : commande ls

La commande ls (list sort = liste triée) affiche le contenu d'un répertoire : fichiers et sous-répertoires. Cette commande supporte de nombreuses options, en particulier des options de tri pour la présentation du résultat. Seules les plus utiles sont listées ci-dessous :

-F	Ajoute un / à la fin du nom des répertoires.
-R	Affiche récursivement le contenu des sous-répertoires. Si un sous répertoire existe, son contenu est affiché.
-1	Affiche des informations très complètes sur les fichiers et répertoires (type, droits d'accès, nom du propriétaire).
-a	Affiche tous les fichiers, y compris ceux qui commencent par un . et qui sont des fichiers cachés.
nom_de_fichier	Nom de fichier à afficher, accepte les "jokers" * et ?.

Pour connaître toutes les options de la commande, taper la commande « man ls »

4. Modifier l'arborescence

4.1. Créer un répertoire

La commande mkdir (make directory) permet de créer un répertoire dans le répertoire courant. La commande est mkdir nom_du_repertoire.

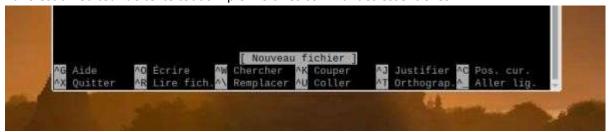
A EXERCICE 5 :

- Déplacez-vous dans le répertoire /home/pi
- Notez ici le contenu de ce répertoire.2
- Créer un répertoire Mes_doc_NSI en utilisant la commande mkdir. Commande tapée :
- Déplacer vous dans le répertoire Mes_doc_NSI . Commande tapée :
- Créez les répertoire suivants : Mes_photos , Mes_doc_pdf , Mes_sites_web
- Taper la commande mkdir -p Mes_algorithmes/programmes . Quel est son effet ?

4.2. Créer un fichier avec un éditeur de texte

EXERCICE 6 :

- Déplacez-vous dans le répertoire /home/pi/ Mes_doc_NSI/ Mes_algorithmes/programmes en utilisant un chemin relatif.
- Tapez la commande nano compteur.py . Vous parvenez alors dans la fenêtre de l'éditeur de texte « nano ».
 nano est un éditeur de texte tout simple. Voici les commandes essentielles :



- Ecrire un programme en langage Python qui initialise un compteur qui compte et affiche les entiers de 0 à 25 inclus.
- Enregistrez votre programme en tapant sur « CTRL+O » puis « Entrée »



- Quitter "nano" : CTRL+X
- Vérifier que votre fichier a été créé dans le répertoire "programme"
- Visualiser le contenu du fichier en tapant more compteur.py . Le conteur s'affiche en lecture seule.
- Exécuter votre programme en ligne de commande grâce à : python3 compteur.py .

4.3. Copier des fichiers

La copie de fichiers ou de répertoires se fait grâce à la commande cp . La syntaxe est la suivante : cp [OPTION] <fichier> <destination>

Vous pouvez accéder aux options de la commande avec : cp - -help .

fichier	Nom du fichier à copier, appelé aussi fichier source.
destination	Si destination est un nom de fichier, le fichier source est recopié dans le fichier destination. Si destination est un nom de répertoire, le fichier source est copié dans ce répertoire.
-i	Cette option indique au système d'interroger l'utilisateur avant d'écraser un fichier existant.
-u	Le système ne fait pas la copie si le fichier destination a une date de dernière modification égale ou plus récente que celle du fichier source.

EXERCICE 7 :

- Créer le répertoire /home/pi/ Mes_doc_NSI/ Mes_algorithmes/programmes_python
- Copier le fichier compteur.py dans le répertoire programmes_python/ et le renommer compteur2.py .
 Quelle commande avez-vous tapé ?
- Modifier le programme compteur2.py afin qu'il compte de 0 à 13 puis de 13 à 0.



PROPRIÉTÉ 3 : Copier un fichier vers un autre fichier :

Si le fichier n'existe pas, le fichier est copié. Si la destination existe, le fichier est écrasé.

EXERCICE 8 :

- Positionnez-vous dans le répertoire /Mes_doc_NSI/Mes_algorithmes/programmes et créez un nouveau fichier nommé remplissLilste.py dans lequel vous taperez un programme permettant de remplir une liste contenant les 10 premières puissances de 2.
- Copiez les deux fichiers compteur.py et remplissListe.py dans
 /Mes_doc_NSI/Mes_algorithmes/programmes_python avec la commande suivante :
 cp -iu fichier1_1 fichier_2 ... fichier_n repertoire .
 Quelle est la commande que vous avez tapé ?

4.4. Copier un répertoire

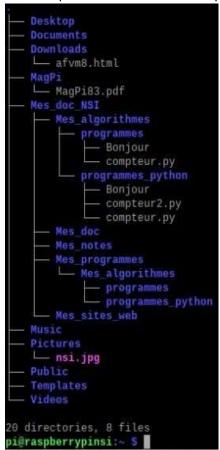
La syntaxe est la suivante : cp [options] <repertoire_source> <repertoire_destination>

fichier	Nom du fichier à copier, appelé aussi fichier source.
destination	Si destination est un nom de fichier, le fichier source est recopié dans le fichier destination. Si destination est un nom de répertoire, le fichier source est copié dans ce répertoire.
-i	Cette option indique au système d'interroger l'utilisateur avant d'écraser un fichier existant.
-u	Le système ne fait pas la copie si le fichier destination a une date de dernière modification égale ou plus récente que celle du fichier source.
-R	Copie récursivement les sous-répertoires du répertoire source. La présence de R lors de la copie de répertoires évite l'affichage d'un message d'erreur.

A EXERCICE 9:

- Au sein du répertoire /home/pi/ Mes doc NSI, créez le répertoire /Mes programmes
- Copiez-le répertoire /Mes_algorithmes ainsi que son arborescence vers le répertoire /home/pi/Mes_doc_NSI/Mes_programmes

- Vérifiez la bonne exécution de l'opération en utilisant une seule commande.
- Vérifiez que votre arborescence depuis Mes_doc_NSI est la suivante :



4.5. <u>Déplacer et renommer un fichier</u>

La syntaxe est la suivante : mv [options] <source> <destination>

<u> </u>		
source	Nom du fichier ou du répertoire source.	
destination	Nom du fichier ou du répertoire destination. Si destination est un nom de répertoire, mv déplace les fichiers sources dans ce répertoire. Si source et destination sont des fichiers dans le même système de fichiers, mv renomme le fichier. Si source et destination sont des fichiers dans des systèmes de fichiers différents, mv déplace le fichier. Si source est un nom de répertoire, mv renomme ce répertoire.	
-f	Force l'écrasement des fichiers existants.	
-i	Interroge l'utilisateur avant d'écraser un fichier. Si -f et -i existent sur la ligne de commande, c'est le dernier qui l'emporte.	
-u	Le système ne fait pas la copie si le fichier destination a une date de dernière modification égale ou plus récente que celle du fichier source.	
-v	Indique le nom des fichiers déplacés sur l'écran.	

EXERCICE 10:

A Positionnez-vous dans le répertoire /home/pi/ Mes_doc_NSI/Mes_programmes

B Déplacez le répertoire Mes_algorithmes/programmes_python vers Mes_programmes en le renommant Mes_programmes/Mes_programmes_python

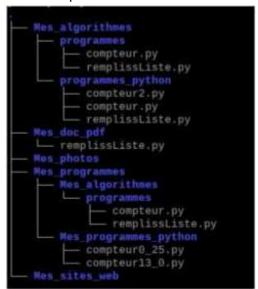
C Vérifiez la bonne exécution de l'opération.

D Positionnez-vous dans /home/pi/ Mes_doc_NSI/Mes_programmes/Mes_programmes_python

E Renommez compteur.py en compteur0 25.py et compteur2.py en compteur0 13 0.py

F Depuis le répertoire /home/pi/ Mes_doc_NSI/Mes_programmes/Mes_programmes_python, déplacez le fichier remplissListe.py vers le répertoire /home/pi/ Mes doc NSI/Mes doc pdf en utilisant un chemin relatif.

G Vérifier que votre arborescence est la suivante :



4.6. Supprimer des fichiers

La syntaxe est la suivante : rm [options] <nom_de_fichier>

nom_de_fichier	Nom du fichier ou du répertoire à effacer.
-f	Ne pas avertir l'utilisateur si un fichier à effacer n'existe pas. Forcer l'effacement sans demander de confirmation si le fichier existe.
-i	Interroge l'utilisateur avant d'effacer un fichier. Si -f et -i existent sur la ligne de commande, c'est le dernier qui l'emporte.
-R	Efface récursivement les sous-répertoires du répertoire source et le répertoire lui- même.
-v	Indique le nom des fichiers effacés sur l'écran.

EXERCICE 11:

A Supprimez le répertoire Mes_doc_NSI/ Mes_algorithmes avec son contenu

B Supprimez le répertoire Mes_doc_NSI/Mes_programmes/Mes_algorithmes avec son contenu

C Supprimez le répertoire Mes_doc_NSI/Mes_programmes/Mes_algorithmes/programmes avec son contenu

D Vérifiez le contenu de votre arborescence. Celle-ci doit être similaire à :

Afficher le contenu d'un fichier

La syntaxe est la suivante : cat [options] <nom_de_fichier_1> <nom_de_fichier_2>

nom_de_fichier_1	Nom du fichier dont l'utilisateur veut visualiser le contenu. Si plusieurs noms de fichiers sont présents dans la ligne de commande, cat affiche leur contenu à l'écran à la suite.
-n	Numérote les lignes affichées sur l'écran. Utile quand il faut retrouver une ligne qui pose problème.

EXERCICE 12: Afficher le contenu des deux programmes compteur0_25.py et compteur0_13_0.py .

5. Gestion des utilisateurs

Les systèmes de type "UNIX" sont des systèmes multi-utilisateurs, plusieurs utilisateurs peuvent donc partager un même ordinateur, chaque utilisateur possédant un environnement de travail qui lui est propre.

Chaque utilisateur possède certains droits lui permettant d'effectuer certaines opérations et pas d'autres.

Un utilisateur un peu particulier est autorisé à modifier tous les droits : ce "super utilisateur" est appelé

"administrateur" ou "root". L'administrateur pourra donc attribuer ou retirer des droits aux autres utilisateurs.

Au lieu de gérer les utilisateurs un par un, il est possible de créer des groupes d'utilisateurs. L'administrateur attribue des droits à un groupe au lieu d'attribuer des droits particuliers à chaque utilisateur.

Les fichiers et les répertoires possèdent 3 types de droits :

- les droits en lecture (symbolisés par la lettre r) : est-il possible de lire le contenu de ce fichier
- les droits en écriture (symbolisés par la lettre w) : est-il possible de modifier le contenu de ce fichier
- les droits en exécution (symbolisés par la lettre x) : est-il possible d'exécuter le contenu de ce fichier (quand le fichier du code exécutable)

Il existe 3 types d'utilisateurs pour un fichier ou un répertoire :

- le propriétaire du fichier (par défaut c'est la personne qui a créé le fichier), il est symbolisé par la lettre u
- un fichier est associé à un groupe, tous les utilisateurs appartenant à ce groupe possèdent des droits particuliers sur ce fichier. Le groupe est symbolisé par la lettre g
- tous les autres utilisateurs (ceux qui ne sont pas le propriétaire du fichier et qui n'appartiennent pas au groupe associé au fichier). Ces utilisateurs sont symbolisés la lettre "o"

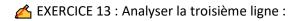
Il est possible d'utiliser la commande "Is" avec l'option "-l" afin d'avoir des informations supplémentaires.

```
david@david-XPS-13-9343 ~/nsi $ ls -l
total 4
-rw-r--r-- 1 david david 0 avril 13 19:58 fic.txt
drwxr-xr-x 2 david david 4096 avril 13 20:05 info
-rw-r--r-- 1 david david 0 avril 12 09:19 photo.jpg
david@david-XPS-13-9343 ~/nsi $ ■
```

✓ EXEMPLE 1 :

- 1. Lisons la première ligne de gauche à droite.
- le premier symbole "-" signifie que l'on a affaire à un fichier, dans le cas d'un répertoire, nous aurions un "d" (voir la 2e ligne)
- les 3 symboles suivants "rw-" donnent les droits du propriétaire du fichier : lecture autorisée (r), écriture autorisée (w), exécution interdite (- à la place de x)
- les 3 symboles suivants "r—"donnent les droits du groupe lié au fichier : lecture autorisée (r), écriture interdite (- à la place de w), exécution interdite (- à la place de x)
- les 3 symboles suivants "r—"donnent les droits des autres utilisateurs : lecture autorisée (r), écriture interdite (- à la place de w), exécution interdite (- à la place de x)
- le caractère suivant "1" donne le nombre de liens (nous n'étudierons pas cette notion ici)
- le premier "david" représente le nom du propriétaire du fichier
- le second "david" représente le nom du groupe lié au fichier
- le "0" représente la taille du fichier en octet (ici notre fichier est vide)
- "avril 13 19 :58" donne la date et l'heure de la dernière modification du fichier "fic.txt" est le nom du fichier ricart.lvh@laposte.net 9/11Chapitre 6 TP UNIX Premiere NSI
- 2. Lisons la deuxième ligne de gauche à droite.
- le premier symbole "d" signifie que l'on a un répertoire
- les 3 symboles suivants "rwx" donnent les droits du propriétaire du répertoire : lecture du contenu du répertoire autorisée (r), modification du contenu du répertoire autorisée (w), il est possible de parcourir le répertoire (voir le contenu du répertoire) (x)

- les 3 symboles suivants "r-x" donnent les droits du groupe lié au répertoire : modification du contenu du répertoire interdite (- à la place de w)
- les 3 symboles suivants "r-x" donnent les droits des autres utilisateurs : modification du contenu du répertoire interdite (- à la place de w)
- le caractère suivant "2" donne le nombre de liens (nous n'étudierons pas cette notion ici)
- le premier "david" représente le nom du propriétaire du répertoire
- le second "david" représente le nom du groupe lié au répertoire
- le "4096" représente la taille du répertoire en octets
- "avril 13 20 :05" donne la date et l'heure de la dernière modification du contenu du répertoire
- "info" est le nom du répertoire



5.1. Changer les droits utilisateurs

Il est important de ne pas perdre de vu que l'utilisateur "root" a la possibilité de modifier les droits de tous les utilisateurs.

Le propriétaire d'un fichier peut modifier les permissions d'un fichier ou d'un répertoire à l'aide de la commande chmod. Pour utiliser cette commande, il est nécessaire de connaître certains symboles :

- les symboles liés aux utilisateurs : "u" correspond au propriétaire, "g" correspond au groupe lié au fichier (ou au répertoire), "o" correspond aux autres utilisateurs et "a" correspond à "tout le monde" (permet de modifier "u", "g" et "o" en même temps)
- les symboles liés à l'ajout ou la suppression des permissions : "+" on ajoute une permission, "-" on supprime une permission, "=" les permissions sont réinitialisées (permissions par défaut)
- $\bullet \ \text{les symboles liés aux permissions}: "r": \text{lecture, "w"}: \text{\'ecriture, "x"}: \text{ex\'ecution}.$

La syntaxe est donc la suivante : chmod [u g o a] [+ - =] [r w x] nom_du_fichier

✓ EXEMPLE 2 : Par exemple la commande chmod o+w toto.txt attribuera la permission "écriture" pour le fichier "toto.txt" "aux autres utilisateurs"

EXERCICE 14:

A. Créer un nouveau fichier dans le répertoire Mes_docs_NSI nommé toto.txt avec la commande touch toto.txt.

B. Vérifier les permissions du fichier créé.
• Propriétaire :
• Groupe :
• Autres :
C. Autoriser les autres à exécuter ce fichier. Commande :
D. Empêcher le groupe de lire ce fichier. Commande :
E. Autoriser tout le monde à écrire dans ce fichier. Commande :
F. Empêcher le propriétaire d'exécuter le fichier. Commande :

6. Gestion des utilisateurs

6.1. Ajouter un utilisateur

Les systèmes de type "UNIX" sont des systèmes multi-utilisateurs, plusieurs utilisateurs peuvent donc partager un même ordinateur, chaque utilisateur possédant un environnement de travail qui lui est propre. Dans l'arborescence actuelle il y a u utilisateur nommé pill est possible d'ajouter un utilisateur.

- 1. Changer le mot de passe root : sudo passwd root et taper un nouveau mot de passe root. Pour que tout le monde ait le même, taper root
- 2. Accéder à la console en mode super-utilisateur : su -
- 3. Ajouter un utilisateur adduser nom utilisateur. Ajouter un utilisateur nommé toto, mot de passe toto
- 4. Vérifier que le nouvel utilisateur a bien été créé en allant dans home
- 5. Déconnectez-vous et logger vous sous le nom toto

6.2. Ajouter un utilisateur à un groupe

- 1. Identifiez-vous avec le compte pi mot de passe raspberry.
- 2. Accéder à la console en mode super-utilisateur : su -
- 3. créer un groupe nommé eleve avec la commande sudo addgroup eleve
- 4. Ajouter toto au groupe eleve avec la commande usermod -aG toto eleve

6.3. Suppression d'un utilisateur

- 1. Accéder à la console en mode super-utilisateur : su -
- 2. supprimer toto du groupe eleve avec la commande sudo deluser toto

6.4. Suppression d'un groupe

- 1. Accéder à la console en mode super-utilisateur : su -
- 2. supprimer le groupe eleve avec la commande sudo delgroup eleve