RAPPORT DE PROJET

Développement d'un système de fichiers et d'une

Distribution Linux personnalisée

Réalisé par :

BOUCHOUATA Hania

ALAOUI MHAMDI Hamza

EL GHAZOUANI Sara

AKHIR Abir

HAOUHAOU Mohammed

Encadré par :

Dr. CHERRADI Mohamed

Sommaire :

Introduction :

Une distribution personnalisée Linux, souvent appelée "custom Linux distribution" en anglais, fait référence à une version spécialement créée et modifiée d'un système d'exploitation Linux existant. Les utilisateurs peuvent personnaliser une distribution Linux en ajustant divers paramètres, en ajoutant ou supprimant des logiciels, en modifiant l'apparence et en intégrant des fonctionnalités spécifiques selon leurs besoins.

Pour créer une distribution personnalisée, les utilisateurs peuvent utiliser des outils comme Linux From Scratch (LFS) qui permettent de construire un système Linux à partir de zéro, ou bien modifier une distribution existante en utilisant des outils tels que Ubuntu Customization Kit (UCK) pour Ubuntu, ou SUSE Studio pour openSUSE.

Une distribution personnalisée Linux est une version adaptée du système d'exploitation Linux, conçue pour répondre aux besoins particuliers d'un utilisateur ou d'une organisation.

Dans chaque distribution Linux, le système de fichiers est une composante fondamentale qui organise la manière dont les données sont stockées, structurées et accessibles. Les distributions Linux utilisent généralement le système de fichiers ext4, mais d'autres systèmes de fichiers tels que Btrfs, XFS, et d'autres peuvent également être utilisés en fonction des besoins.

Lors de la création d'une distribution personnalisée Linux, les utilisateurs peuvent choisir le système de fichiers qui convient le mieux à leurs besoins spécifiques en termes de performances, de sécurité et de fonctionnalités. Les options de configuration du système de fichiers, les points de montage, et d'autres paramètres associés au stockage des données peuvent être

personnalisés pour répondre aux exigences particulières de l'utilisateur ou de l'organisation.

2- Système de fichier :

2-1- le langage de programmation :

Le système d'exploitation linux est basé sur le langage de programmation c. Inventé au début des années 1970 pour réécrire UNIX, C'est devenu un des langages les plus utilisés. Il est qualifié de langage de base niveau dans le sens où chaque instruction du langage est conçue pour être compilée en un nombre d'instruction machine assez prévisible en termes d'occupation mémoire et de charge de calcul. Ces caractéristiques en font un langage privilégié quand on cherche à maîtriser les ressources matérielles utilisées, le langage machine et les données binaires générées par les compilateurs étant relativement prévisibles. Ce langage est donc extrêmement utilisé dans des domaines comme la programmation embarquée sur microcontrôleurs les calculs intensifs, l'écriture de systèmes d'exploitation et les modules où la rapidité de traitement est importante.

Pour cela on va utilisons le langage de programmation c dans les fonctions et les commandes de pour gère notre système de fichier.

2-2-Les bibliothèque pour le codage :

Pour gérer les fichiers et les répertoires dans notre système de fichiers. le langage c offre plusieurs bibliothèques pour exécutée des commandes et naviguée dans le système de fichiers.

Les bibliothèques des commandes de projet :

<stdio.h> : Fournit des fonctions pour les opérations d'entrée/sortie standard, telles que printf et scanf.

<stdlib.h> : Contient des fonctions liées à la gestion de la mémoire dynamique, aux conversions de type, aux fonctions de tri, etc. Des fonctions telles que malloc, free, et exit sont incluses.

<unistd.h> : Fournit un accès à des fonctions liées à l'environnement du système d'exploitation Unix. Cela inclut des appels système tels que fork (pour créer des processus), exec (pour exécuter des programmes), et getpid (pour obtenir l'identifiant du processus courant).

<sys/stat.h> : Fournit des déclarations de fonctions et de structures utilisées pour l'accès et la manipulation des attributs de fichier. Inclut des fonctions comme mkdir (pour créer des répertoires) et des structures comme struct stat pour stocker des informations sur les fichiers.

<string.h> : Cette bibliothèque offre des manipulations de chaînes de caractères.

<pwd.h> : offre des structures et des fonctions pour accéder à la base de do
<grp.h> : offre des fonctionnalités liées à la gestion des groupes d'utilisateurs sur les systèmes Unix.

<sys/types.h> : fournit des déclarations de types de données utilisées dans des opérations système spécifiques.

<dirent.h> : fournit des déclarations et des fonctions pour la manipulation de répertoires et la lecture des entrées de répertoire.

2-3- Les fonctions et les commande :

Les commandes intégrées dans le système de fichiers :

delete : fonction delete permet de supprimer un ou plusieurs fichiers avec l'option " -f " et

Les dossiers avec les deux " -d ".

create : permet de crée des fichiers avec l'option " -f " et des répertoires avec l'option " -d " cette commande permet aussi de créer plusieurs fichiers ou plusieurs dossiers en une meme instruction.

Copy : permet de copier le contenue d'un autre fichier.

list : lister le contenue d'un répertoire.

Rights : permet de gérer les droits des utilisateurs.

Show : afficher le contenu d'un fichier spécifié.

Currentdir: la fonction principale qui affiche le chemin du répertoire courant.

searsh: donne les fichiers qui commencent par une chaine de caractères donnes

chowner: Elle permet aux utilisateurs de changer la propriété de l'utilisateur et du groupe à la fois pour les fichiers et pour le répertoire.

Move : propose une interface en ligne de commande permettant de déplacer ou renommer Un fichier, on a utilisé la fonction Rename pour effectuer l'opération de renommage et de déplacement.

Clean : pour effacer l'écran de la console en utilisant des codes d'échappement ANSI. Les codes d'échappement \033[H\033[J représentent les séquences ANSI pour déplacer le curseur à la position de départ et effacer l'écran.

compress: Elle lit un fichier source, compte le nombre de caractères consécutifs identiques, et écrit ces informations dans un fichier de destination.

Removeacc: supprimer des utilisateurs sur un système Linux en utilisant la commande userdel. Il prend en charge la suppression de plusieurs utilisateurs en utilisant une boucle sur les arguments fournis en ligne de commande. Le processus de suppression implique la vérification de l'existence de l'utilisateur, la construction d'une commande pour supprimer l'utilisateur avec userdel -r, et l'exécution de cette commande avec system. L'utilisation de userdel -r supprime également le répertoire personnel de l'utilisateur et tous ses fichiers.

3- Création de la distribution linux :

3-1- Les méthode de création de distribution :

Parmi les méthodes de construction de distribution linux qu'on a trouvé, il y a 3 méthodes principales qu'on a testé :

- La méthode de création de distribution à base :

Dans ce cas on a utilisé Arch linux (une distribution linux qui constitue une catégorie elle-même), alors que notre distribution sera basée sur Arch linux.

Principalement, on a utilisé un profile (releng) parmi les deux profiles valables sur le repo Gitlab de l'outil de création de distributions "archiso" : https://gitlab.archlinux.org/archlinux/archiso (pour plus d'informations il existe une documentation complète le "archwiki" : https://wiki.archlinux.org/title/archiso).

On a implémenté notre configuration dans le profile et on a ajouté un "installer" qui est "Calamares" (pour plus d'info : https://calamares.io/).

Enfin on a transformé notre profile en image iso en utilisant la commande mkarchiso (cette commande est valable seulement après installer l'outil archiso soit par la commande "*sudo pacman –S archiso*", ou directement par la compilation en utilisant le makefile present sur le repo gitlab "*make install*"). La commande pour la transformation du profile en image iso est: "*sudo mkarchiso –v –w /path/to/work/directory -o /path/to/output/directory /path/to/profile*" et on trouve que notre image iso est présente dans le chemin /path/to/output/directory

NB : on n'a pas pu créer une image iso qui est utilisable, malgré qu'on ait investi beaucoup de temps par cette méthode.

- La méthode de modification une version minimale :

On a essayé d'utiliser une version server minimal d'une distribution et on a construit notre distribution à partir de la dernière (installer un window manager, un login manager, desktop...).

Apres qu'on a fait cela, on passe vers la partie de création de l'image iso en utilisant un system backup ou monter le disque dur de la distribution sur une distribution live (la distribution doit être éteinte) puis utiliser un outil de création d'image iso pour créer l'image.

- La méthode de modification d'une image iso déjà existante :

Cette méthode consiste à modifier une image iso en utilisant un programme comme "Cubic" pour les distributions basées sur Ubuntu et suivre les étapes jusqu'à obtenir notre propre image iso.

En générale, cette méthode n'est pas conseillée parce qu'elle modifie seulement la version live de l'image iso et ne crie pas des modifications permanentes au cas de l'installation totale de l'image.

3-2- La méthode utilisé :

On a décidé d'utiliser la 2ème méthode, on a utilisé Ubuntu server comme base et on a passé par les étapes suivantes pour créer notre propre distribution :

- Installer un Desktop Environment:

  ```
  sudo apt install xfce4
  ```

- Installer et configurer un display manager (login manager) :

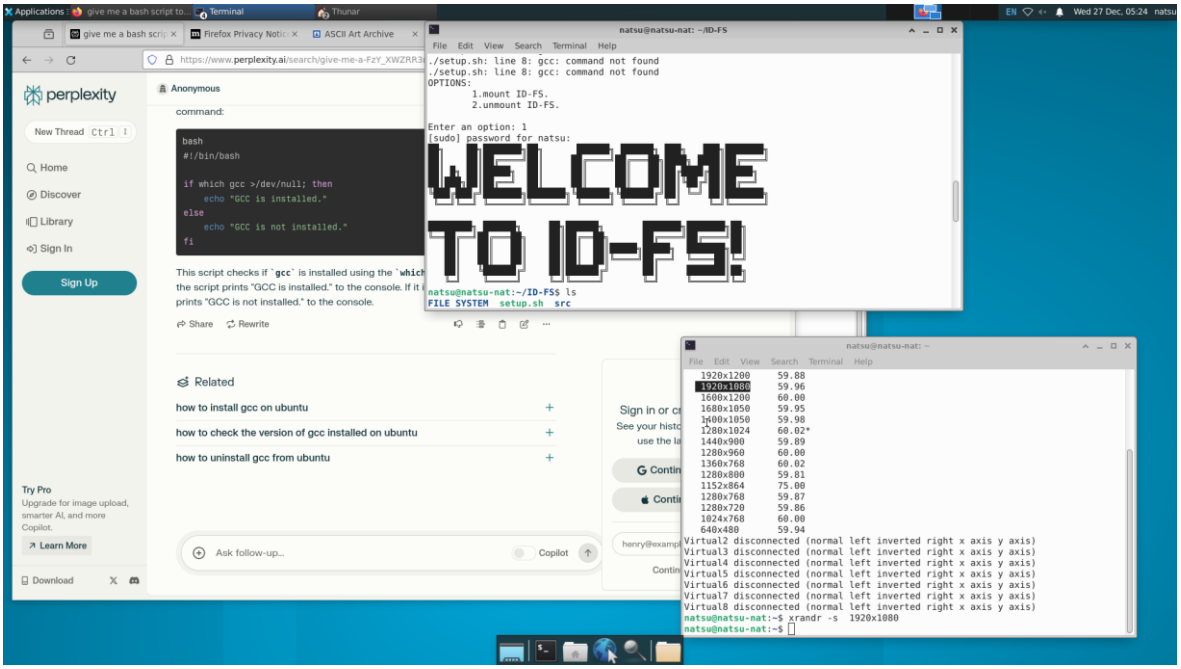
  sudo apt install sddm

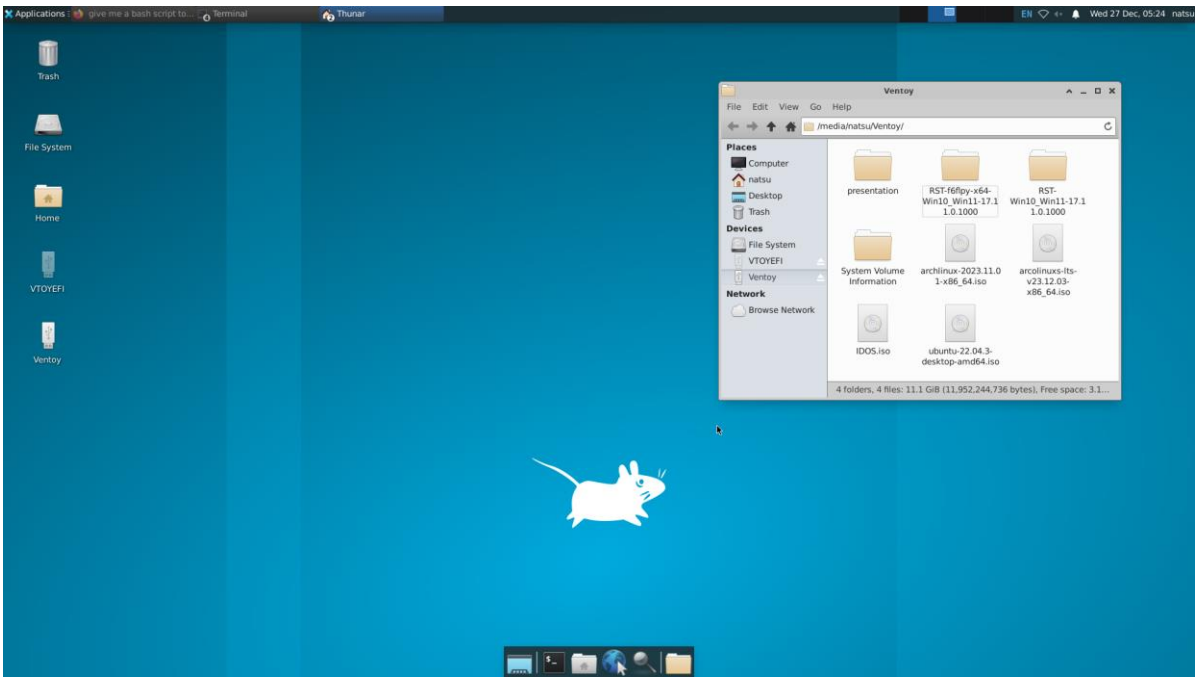
  sudo dpkg-reconfigure sddm

  sudo systemctl enable sddm

  sudo systemctl start sddm

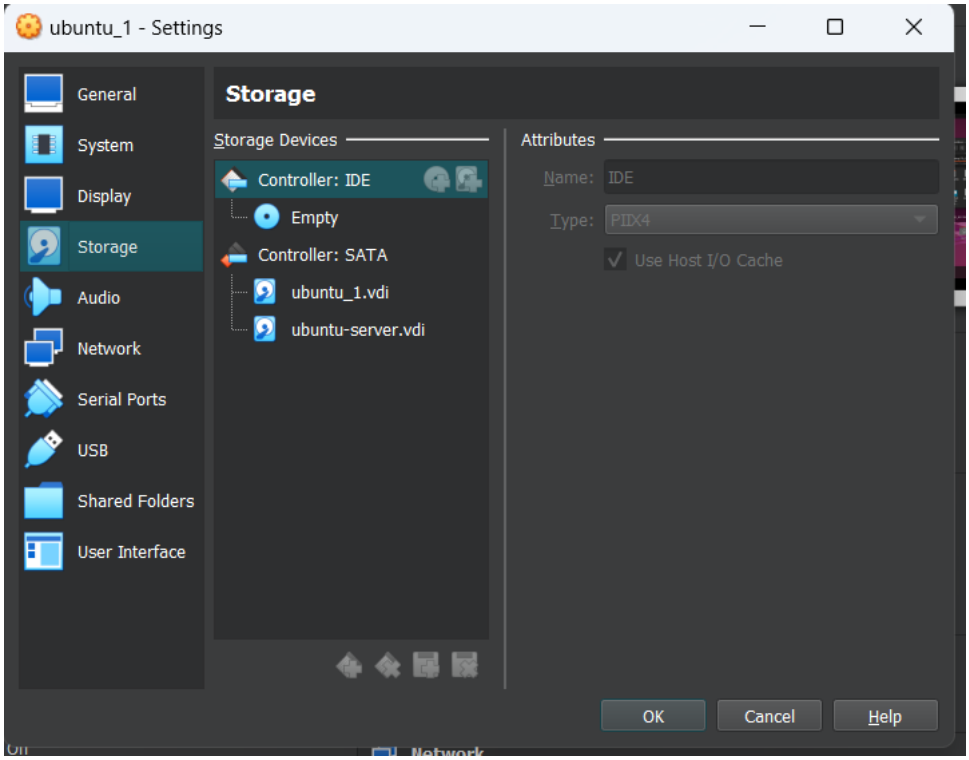
- On redémarre la distribution pour voir le résultat qui est dans notre cas :





Après cette étape on passe vers l'étape de transformation en image iso en utilisant l'outil "genisoimage" en passant par les étapes suivantes :

- Sur Virtual Box on arrête notre distribution.
- On ajoute le disque de notre distribution dans la liste des disques d'une autre distribution (par exemple Ubuntu).



- On démarre l'autre distribution :
- On monte le disque de notre distribution de la manière suivante :

```
sudo mkdir -p /mnt/idos
sudo mount /dev/dm-0 /mnt/idos
```

- On utilise "genisoimage":

```
Sudo genisoimage -o ~/ID_DISTRO.iso -R -J -joliet-long ~/mnt/idos
```

- On obtient le résultat suivant :

```
 99.86% done, estimate finish Mon Jan  8 01:58:19 2024
 99.99% done, estimate finish Mon Jan  8 01:58:19 2024
Total translation table size: 0
Total rockridge attributes bytes: 16306445
Total directory bytes: 42917232
Path table size(bytes): 183094
Max brk space used 892f000
3840333 extents written (7500 MB)
natsu@natsu-VirtualBox:~$ ls -l ID_DISTRO.iso
-rw-r--r-- 1 root root 7865001984 Jan  8 01:58 ID_DISTRO.iso
natsu@natsu-VirtualBox:~$
```

Et voilà, on est arrivé à créer notre image iso et on peut l’installer.

Conclusion :

Ce projet a été une occasion d'approfondir les connaissances sur les systèmes d'exploitation, la programmation en C et la personnalisation des distributions Linux. Le choix du langage de programmation C a été délibéré en raison de sa performance et de sa prédictibilité. Les bibliothèques standard et les commandes du système de fichiers ont été utilisées pour mettre en œuvre des fonctionnalités essentielles. La méthode de création de la distribution personnalisée s'est concentrée sur la sauvegarde du système, offrant une image ISO pour une installation facile sur diverses machines.