**Université Abdelmalek Essaadi**

Ecole Nationale des Sciences Appliquées Al-Hoceima



Rapport du Projet

Première Année Ingénierie des données

**Réalisé par :**

|  |
| --- |
| * Kawtar BENALI |
| * Oumaima OUBAHA |
| * Salma CHAKOURI |
| * Mozdalifa AZDOUD |

**Encadré par :** Pr. Mohamed CHERRADI

Année Universitaire : 2023/2024

**Remerciement**

Nous tenons à exprimer notre profonde gratitude envers notre enseignant, qui nous a confié le passionnant projet d'Ingénierie de Données 1 (ID1FS). En qualité d'étudiants en ingénierie de données, cette mission a représenté bien plus qu'un simple exercice académique ; elle a été une immersion captivante dans le monde complexe et en constante évolution de la gestion des données.

Nous sommes reconnaissants envers notre professeur pour sa confiance en notre capacité à relever les défis de ce projet ambitieux. Ses conseils éclairés, sa passion communicative pour le domaine de l'ingénierie de données et son soutien constant ont joué un rôle essentiel dans la réussite de notre projet. Cette opportunité unique nous a permis de mettre en pratique nos connaissances théoriques tout en développant des compétences cruciales pour notre future carrière en tant qu'ingénieurs de données.

Nos remerciements sincères vont à notre professeur pour avoir guidé ce projet avec enthousiasme et expertise, contribuant ainsi de manière significative à notre formation professionnelle. Cette expérience restera un jalon mémorable dans notre parcours académique et constituera une base solide pour nos futures entreprises dans le domaine de l'ingénierie de données.

Nous tenons à exprimer notre gratitude pour cette opportunité enrichissante et pour l'engagement continu de notre enseignant envers notre succès en tant qu'étudiants en ingénierie de données. Ces moments de collaboration et d'apprentissage demeureront gravés dans nos souvenirs académiques et constitueront une source d'inspiration pour les défis à venir. Merci infiniment pour cette expérience exceptionn

**Résumer**

Le projet ID1FS, conçu dans le cadre de notre cursus en ingénierie de données, a représenté une exploration immersive du monde complexe de la gestion de données. Supervisés par notre enseignant dévoué, nous avons démarré le projet en personnalisant une distribution Linux, ID1FS, selon des spécifications précises. En utilisant l'outil Cubic, nous avons ajusté l'image Ubuntu de base, intégrant des fonctionnalités uniques et une identité visuelle distinctive.

Une étape cruciale a été la création d'un système de fichiers robuste, ID1FS, avec un mécanisme de journalisation exhaustif. La gestion minutieuse des permissions des fichiers a été un pilier de la sécurité du système, offrant une granularité précise dans le contrôle des actions autorisées. Des commandes améliorées ont été introduites pour simplifier l'interaction avec le système de fichiers.

L'expérience utilisateur a été optimisée par la personnalisation du terminal ID1FS, permettant aux utilisateurs de se connecter, de s'inscrire, et d'interagir intuitivement avec le système. La distribution a également bénéficié d'une identité visuelle forte grâce à l'intégration de GNOME TWEAKS, offrant des thèmes visuels personnalisables.

La mise en œuvre du projet a nécessité une installation minutieuse de l'image personnalisée sur une machine virtuelle Windows, offrant un environnement dédié pour des tests approfondis du système ID1FS. Les étapes du processus, depuis la personnalisation jusqu'à l'installation sur la machine virtuelle, ont été soigneusement documentées dans un guide d'installation complet.

Table des matières

[I- Introduction 2](#_Toc90723559)

[**1.** **La création d’un système de fichier personnalisé** 3](#_Toc90723560)

[**2.** **La personnalisation d’une distribution ubuntu** 3](#_Toc90723560)

[II- Système de fichier 4](#_Toc90723564)

[**1.** **Outils et méthode** 3](#_Toc90723563)

[**2.** **Hiérarchie** 3](#_Toc90723563)

[**3.** **Le hachage** 3](#_Toc90723563)

[**4.** **Loging** 3](#_Toc90723563)

[**5.** **Les permissions**  3](#_Toc90723563)

[**6.** **Les commandes**  3](#_Toc90723563)

[**7.** **La méthadonnée** 3](#_Toc90723563)

[IIl- Distribution personnalisé 4](#_Toc90723564)

[**1.** **Définition cubic** 4](#_Toc90723565)

[**2.** **Personnalisation de la distribution linux avec ubuntu et cubic** 4](#_Toc90723565)

[**3.** **Design personnalisé de la distribution ubuntu ’’ID1FS’’** 4](#_Toc90723566)

[**4.** **Fonctionnement de la distribution** 4](#_Toc90723567)

1. **Introduction**

Au cœur de l'innovation technologique, notre projet se démarque par une vision audacieuse axée sur la création d'un système de fichiers personnalisé et la personnalisation approfondie d'une distribution Linux basée sur Ubuntu. S'inscrivant dans le domaine en constante évolution de l'informatique, notre initiative vise à repousser les frontières de la personnalisation des systèmes d'exploitation, offrant ainsi une expérience utilisateur unique et adaptée aux besoins spécifiques.

Cette exploration multidimensionnelle combine la conception minutieuse d'un système de fichiers innovant avec la personnalisation poussée d'une distribution Ubuntu. Notre rapport plonge dans les détails de cette entreprise technologique, mettant en lumière les choix stratégiques, les défis relevés et les solutions créatives qui ont donné forme à notre projet.

Par le biais de cette aventure, nous aspirons à redéfinir les normes de l'environnement Linux en offrant aux utilisateurs finaux une expérience exceptionnelle et sur mesure. Suivez-nous à travers ce rapport captivant qui dévoile les coulisses de notre projet, révélant comment chaque décision, chaque ligne de code contribue à l'émergence d'un écosystème informatique innovant et adaptatif.

Le projet que nous présentons se distingue par la focalisation sur deux axes fondamentaux qui contribuent à son originalité et à son impact significatif. Ces axes sont les suivants :

1. La création d’un système de fichier personnalisé

La première étape cruciale de notre projet se concentre sur l'établissement d'un système de fichiers novateur et efficace. Nous ambitionnons de concevoir une structure hiérarchique personnalisée, permettant à l'utilisateur d'interagir avec le système via des commandes spécifiques. L'objectif sous-jacent est la création d'un système de fichiers capable de supporter une personnalisation approfondie, tout en garantissant la stabilité et la sécurité du système. Nous sommes déterminés à offrir aux utilisateurs une expérience unique et adaptée à leurs besoins, tout en maintenant les normes élevées de fiabilité et de sécurité.

Dans cette phase du projet, nous chercherons à élargir notre système de fichiers en incorporant des fonctionnalités avancées tout en veillant à ce que l'interface utilisateur demeure intuitive. L'interaction avec le système à travers des commandes spécifiques sera optimisée pour assurer une expérience utilisateur fluide et efficace. Nous nous attacherons également à explorer des mécanismes de personnalisation plus poussés, garantissant ainsi une flexibilité maximale pour les utilisateurs finaux.

2. La personnalisation d’une distribution Ubuntu

La deuxième phase du projet est consacrée à la création d'une distribution Linux sur mesure, basée sur la populaire plateforme Ubuntu. Cette distribution sera conçue en tenant compte des besoins spécifiques de l'utilisateur final. Nous nous pencherons sur divers aspects, tels que les logiciels préinstallés, les paramètres système et l'interface utilisateur, avec une approche résolument orientée vers la personnalisation et l'adaptabilité.

Notre objectif est de fournir une expérience utilisateur unique et optimale en offrant une distribution Ubuntu entièrement adaptée aux exigences individuelles. Cela inclut la sélection minutieuse des logiciels préinstallés, la configuration avancée des paramètres système pour maximiser la performance, et la conception d'une interface utilisateur intuitive et esthétiquement plaisante. Nous mettrons en œuvre des mécanismes flexibles permettant aux utilisateurs de personnaliser leur environnement de travail, renforçant ainsi l'approche centrée sur l'utilisateur de notre projet.

1. **Système de fichier**

Un système de fichiers, également appelé système de gestion de fichiers (SGF), est une structure organisationnelle utilisée par un système informatique pour stocker, organiser et gérer des fichiers et des répertoires. Il fournit une méthode permettant d'accéder, de stocker et de récupérer des données de manière ordonnée et structurée. Un système de fichiers définit la manière dont les données sont nommées, organisées, stockées et récupérées sur un dispositif de stockage, tel qu'un disque dur, une clé USB ou une carte mémoire. Il comprend des éléments tels que des répertoires (ou dossiers), qui peuvent contenir des fichiers ou d'autres répertoires, ainsi que des métadonnées associées à chaque fichier, telles que la date de création, la date de modification, les autorisations d'accès, etc. L'organisation hiérarchique des fichiers et des répertoires dans un système de fichiers permet aux utilisateurs et aux applications d'accéder rapidement aux données souhaitées. Les systèmes de fichiers jouent un rôle crucial dans la gestion et la structuration des informations stockées sur un support de stockage, facilitant ainsi le stockage, la recherche, la modification et la suppression de données de manière efficace.

1 Outils et méthode

Pour donner vie à notre système de fichiers, nous avons fait appel à un ensemble d'outils et de bibliothèques soigneusement sélectionnés, mettant l'accent sur la flexibilité et la sécurité. Nous avons opté pour Python comme langage de développement en raison de sa polyvalence et de sa facilité d'utilisation. Pour optimiser notre environnement de développement, PyCharm a été l'allié idéal, offrant un cadre de travail intuitif et efficace.

En ce qui concerne les bibliothèques, nous avons fait un choix stratégique pour assurer la robustesse et l'efficacité de notre système :

**OS** : Cette bibliothèque facilite l'interaction avec le système d'exploitation, permettant ainsi d'effectuer des opérations sur les fichiers, les répertoires, etc. Elle constitue un pilier fondamental pour la manipulation des ressources système, assurant une compatibilité et une stabilité maximales.

**SHUTIL** : Utilisée pour des opérations de haut niveau sur les fichiers et les répertoires, SHUTIL simplifie les manipulations de manière efficace. Son intégration renforce la flexibilité de notre système en permettant des actions avancées sur les données stockées.

**LOGGING** : Intégré pour la gestion détaillée des journaux, LOGGING offre une visibilité accrue pour le suivi et le débogage du système. Cette fonctionnalité est cruciale pour assurer la traçabilité des opérations et faciliter la résolution rapide des problèmes éventuels.

**HASHLIB** : Employé pour garantir l'intégrité des données à travers des fonctions de hachage sécurisées, HASHLIB renforce la sécurité du système. Cette mesure de sécurité essentielle assure la confidentialité et l'authenticité des informations stockées.

**SUBPROCESS** : Ce module permet d'interagir avec des processus système externes depuis Python, utilisé pour lancer des commandes, récupérer leur sortie et gérer leurs entrées/sorties. Cette fonctionnalité s'avère cruciale pour la communication efficace avec d'autres composants du système.

**JSON** : Utilisé pour la gestion des configurations, JSON facilite le stockage et la récupération des paramètres du système sous forme de fichiers JSON. Cette approche flexible permet aux utilisateurs de personnaliser davantage leur expérience.

**GETPASS** : Intégré pour obtenir le nom d'utilisateur de l'utilisateur actuellement connecté, GETPASS renforce la sécurité en garantissant une authentification précise des utilisateurs.

**PATH** : Cette bibliothèque permet une gestion efficace des chemins d'accès, facilitant la manipulation des emplacements des fichiers et des répertoires. Son utilisation contribue à une organisation claire et structurée des données au sein du système.

**DATETIME** : Utilisé pour la gestion des horodatages, DATETIME contribue à la création d'une structure temporelle organisée au sein du système de fichiers. Cette fonctionnalité est particulièrement utile pour suivre et organiser les modifications et les ajouts de fichiers dans le temps.

2 Hiérarchie

Au cœur de notre système de fichiers personnalisé, nous avons établi une hiérarchie simple et organisée autour du répertoire central "**MyFileSystem**". Cette structure forme la base de notre distribution, en fournissant un cadre efficace pour la gestion des données et des opérations système.

**Logs** : Ce répertoire est dédié à la journalisation des activités. À l'intérieur, le fichier "File\_System.log" enregistre de manière détaillée chaque action, offrant une trace complète pour le suivi et l'analyse.

**Users.json** : Le fichier "users.json" stocke les noms d'utilisateurs et leurs mots de passe hachés. Cette centralisation facilite la gestion sécurisée des identifiants d'utilisateur, contribuant ainsi à la sécurité globale du système.

**Personal\_Folder** : Ce répertoire central héberge plusieurs sous-répertoires essentiels :

**Moved\_File** : Les fichiers temporairement supprimés sont déplacés vers ce répertoire. Il agit comme une zone de rétention avant une éventuelle suppression définitive.

**UserFiles** : Lorsqu'aucun chemin spécifique n'est spécifié pour un fichier, ils sont automatiquement enregistrés dans ce répertoire par défaut. Cela assure une organisation claire et évite la dispersion non planifiée des fichiers.

Directories : Lorsqu'aucun chemin spécifique n'est spécifié pour un répertoire, ils sont automatiquement enregistrés dans Personal\_Folder par défaut.

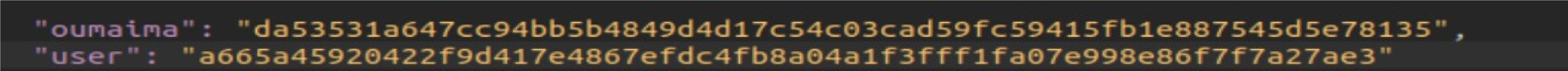
Cette structure garantit une organisation systématique des données, favorisant une gestion efficace tout en offrant des fonctionnalités cruciales telles que la journalisation détaillée des activités et la sécurité renforcée des informations d'identification des utilisateurs.

3 Le hachage

La sécurité des données est une priorité centrale dans notre projet "ID1FS". Afin de garantir la confidentialité des informations d'identification, une approche de hachage a été implémentée pour les mots de passe. Cette technique de sécurité transforme le mot de passe en une série de caractères alphanumériques apparemment aléatoires, offrant ainsi une protection robuste contre toute tentative de lecture visuelle.

Lorsqu'un utilisateur saisit son mot de passe, le système le convertit immédiatement en une empreinte numérique via une fonction de hachage, en l'occurrence la bibliothèque hashlib de Python. Cette empreinte, qui est unique pour chaque mot de passe, est ce qui est réellement stocké dans le système. Ainsi, même en cas d'accès non autorisé aux fichiers de stockage, les mots de passe réels restent indéchiffrables.

Pour assurer une traçabilité, les noms d'utilisateur associés aux mots de passe hachés sont enregistrés dans un fichier nommé "users.json". Ce fichier offre une correspondance entre les utilisateurs et leurs empreintes de mots de passe, facilitant l'authentification ultérieure sans compromettre la sécurité des informations sensibles.



4 Login

Au sein du projet "ID1FS", une attention particulière a été accordée à la mise en place d'un système de logging exhaustif, visant à consigner de manière méthodique chaque action entreprise dans le terminal. Cette démarche vise à garantir une traçabilité complète des opérations effectuées, offrant ainsi un historique détaillé du fonctionnement du système.

Pour centraliser ces informations cruciales, un fichier spécifique baptisé "filesysteme.log" a été créé et est soigneusement conservé dans le répertoire dédié appelé "logs". Ce fichier joue un rôle essentiel en regroupant les détails de chaque interaction au sein du terminal, allant de la création de répertoires à la manipulation de fichiers, en passant par l'exécution de commandes spécifiques.

L'approche méticuleuse de journalisation adoptée permet non seulement de suivre de près les opérations réalisées, mais accorde également une importance particulière à la chronologie de chaque action. Ainsi, chaque entrée dans "filesysteme.log" est marquée avec précision par la date et l'heure de son exécution, facilitant ainsi le suivi temporel et la compréhension de l'évolution du système.

L'implémentation de cette fonctionnalité a été réalisée avec l'utilisation de la bibliothèque login de Python, assurant une robustesse et une fiabilité accrues. Chaque ligne de log consigne des détails tels que l'heure précise de l'action, le type d'opération effectuée, ainsi que toute information pertinente liée à l'événement en question.

5 Les permissions

La gestion minutieuse des permissions des fichiers dans "ID1FS" constitue un pilier fondamental de la sécurité du système. Nous avons choisi une approche simplifiée et explicite pour attribuer des droits à chaque fichier, en utilisant les lettres "R" (Read) pour la lecture, "W" (Write) pour l'écriture, et "X" (Execute) pour l'exécution.

Cette approche offre une granularité précise dans le contrôle des actions autorisées sur les fichiers, permettant à l'utilisateur de déterminer avec finesse qui peut lire, écrire ou exécuter chaque fichier en fonction de ces attributs. Par exemple, un fichier peut être configuré pour autoriser la lecture par tous les utilisateurs, mais limiter l'écriture et l'exécution à un groupe spécifique.

Pour simplifier davantage la gestion des permissions, nous avons introduit la notation "A" qui rassemble les trois autorisations, permettant une application rapide d'autorisations complètes à un fichier. Cette flexibilité dans la définition des permissions offre un contrôle fin sur l'accès aux données, contribuant ainsi à renforcer la sécurité du système.

En mettant l'accent sur la simplicité et la précision, la gestion des permissions dans "ID1FS" offre à l'utilisateur une approche intuitive pour définir et contrôler les droits d'accès aux fichiers. Cette flexibilité permet de configurer de manière personnalisée les droits d'accès, renforçant la sécurité globale du système en assurant une protection robuste contre les accès non autorisés et en préservant la confidentialité et l'intégrité des données stockées.

6 La métadonnée

La commande "**BAYANATE NOM\_DE\_FICHIER**" occupe une position centrale au sein de notre système de fichiers, jouant un rôle essentiel en offrant aux utilisateurs un aperçu détaillé des métadonnées associées à un fichier spécifique. Lorsque cette commande est invoquée, elle déclenche l'affichage d'un ensemble d'informations cruciales qui fournissent une vision exhaustive du fichier en question.

Les métadonnées présentées par "**BAYANATE NOM\_DE\_FICHIER**" sont variées et complètes, comprenant des détails tels que le nom du fichier, son type (extension), la taille en octets, la date de création, ainsi que la date de la dernière modification. Cette compilation riche en informations offre à l'utilisateur une visibilité immédiate sur les caractéristiques essentielles du fichier, permettant ainsi une gestion plus informée et précise au sein du système de fichiers.

En exploitant cette commande, les utilisateurs peuvent rapidement accéder à des détails cruciaux, comme la taille du fichier, sa date de création et de dernière modification. Cette fonctionnalité facilite une évaluation rapide et complète des propriétés de chaque fichier, contribuant ainsi à une gestion plus efficiente de l'espace de stockage et des données.

L'avantage de cette commande réside dans sa capacité à fournir une vue détaillée des fichiers, permettant aux utilisateurs d'obtenir des informations pertinentes sans avoir à naviguer à travers une multitude de menus. Cette approche favorise une gestion plus proactive et éclairée des fichiers, renforçant ainsi la convivialité et la fonctionnalité globale de notre système de fichiers.

7 les commandes

Le Terminal ID1FS, dédié à la gestion de fichiers personnalisée, offre une interface avancée pour simplifier les opérations courantes liées aux fichiers et aux répertoires. Ce terminal, conçu avec une approche centrée sur l'utilisateur, propose une liste étendue de commandes améliorées visant à optimiser la manipulation des données. Que vous soyez novice ou expert en gestion de fichiers, le Terminal ID1FS a été élaboré pour offrir une expérience utilisateur intuitive et efficace.

L'ensemble diversifié de commandes vous permettra de créer, déplacer, copier, supprimer et gérer les autorisations de fichiers et de répertoires de manière personnalisée. Chaque commande a été pensée pour offrir une solution précise à des besoins spécifiques, contribuant ainsi à une gestion systématique et fluide de vos données.

Découvrez ci-dessous la liste complète des commandes disponibles dans le Terminal ID1FS, accompagnée de leurs rôles respectifs et de leur syntaxe. Explorez ces fonctionnalités pour tirer pleinement parti de ce système de gestion de fichiers personnalisé, et n'hésitez pas à intégrer ces commandes dans votre flux de travail quotidien pour une expérience de gestion de fichiers optimale.

**adefr** : Ajouter un répertoire

Syntaxe : adefr <nom\_du\_répertoire>

**adeff** : Ajouter un fichier

Syntaxe : adeff <nom\_du\_fichier> <contenu\_du\_fichier>

**sard** : Liste le contenu d'un répertoire

Syntaxe : sard <répertoire\_à\_lister> ou sard

**azelft** : Supprimer temporairement un fichier

Syntaxe : azelft <nom\_du\_fichier>

**azelfd** : Supprimer un fichier définitivement

Syntaxe : azelfd <nom\_du\_fichier>

**azelr** : Supprimer un répertoire

Syntaxe : azelr <nom\_du\_répertoire>

**rec** : Restaurer un fichier

Syntaxe : rec <fichier\_à\_restaurer>

**adef\_f** : Ajouter un fichier à une prévisualisation

Syntaxe : adef\_f <nom\_du\_fichier> <répertoire\_destination>

**adef\_r** : Ajouter un répertoire à une prévisualisation

Syntaxe : adef\_r <nom\_du\_répertoire> <répertoire\_destination>

**amer** : Définir les permissions

Syntaxe : amer <fichier\_cible> <permissions>

**ana** : Obtenir le nom d'utilisateur

Syntaxe : ana

**badel** : Changer le nom d'un fichier

Syntaxe : badel <nom\_actuel> <nouveau\_nom>

**waqt** : Obtenir la date actuelle

Syntaxe : waqt

**qas** : Couper un fichier

Syntaxe : qas <nom\_du\_fichier> <répertoire\_source> <répertoire\_destination>

**nasq** : Copier un fichier

Syntaxe : nasq <nom\_du\_fichier> <répertoire\_source> <répertoire\_destination>

**baht** : Rechercher un fichier

Syntaxe : baht <nom\_du\_fichier>

**iqra** : Lire le contenu d'un fichier

Syntaxe : iqra <nom\_du\_fichier> [répertoire\_source] ou iqra <nom\_du\_fichier>

**zed** : Ajouter du texte à un fichier

Syntaxe : zed

**bayanat** : Lister les métadonnées d'un fichier

Syntaxe : bayanat

**showlog** : Lire le contenu du fichier journal

Syntaxe : showlog

**exit** : Quitter le terminal ID1FS

Syntaxe : exit

1. **Distribution personnalisée**

Une distribution Linux, souvent appelée distribution Linux ou distro, est une version complète du système d'exploitation Linux, regroupant le noyau Linux, des utilitaires système, des bibliothèques logicielles et souvent un système de gestion de paquets. Ces distributions sont créées par des organisations, des communautés open source ou des entreprises, et elles visent à fournir une solution complète et fonctionnelle pour les utilisateurs de Linux.

Chaque distribution Linux peut avoir des objectifs spécifiques, des caractéristiques uniques, et une philosophie particulière quant à la manière dont elle gère les logiciels, les mises à jour et les configurations système. Certaines distributions sont conçues pour la stabilité et l'utilisation à long terme, tandis que d'autres se concentrent sur l'innovation, l'adoption rapide des dernières technologies, ou des objectifs spécifiques tels que la sécurité, la convivialité, ou le support matériel étendu.

Les distributions Linux populaires incluent Ubuntu, Fedora, Debian, CentOS, Arch Linux, et bien d'autres. Chacune de ces distributions offre une expérience utilisateur unique et peut être adaptée aux besoins spécifiques de l'utilisateur ou de l'entreprise.

La diversité des distributions Linux permet aux utilisateurs de choisir celle qui correspond le mieux à leurs besoins et à leurs préférences, offrant ainsi une flexibilité et une personnalisation qui font la force de l'écosystème Linux.

1 Définition de cubic

Cubic, également connu sous le nom de "Custom Ubuntu ISO Creator," est un outil logiciel sophistiqué qui sert de facilitateur majeur dans le processus de personnalisation d'images ISO d'Ubuntu. En effet, il offre une interface utilisateur graphique conviviale qui permet aux utilisateurs de manière intuitive et accessible d'apporter des modifications approfondies à l'installation standard d'Ubuntu.

L'objectif principal de Cubic est de donner aux utilisateurs un contrôle étendu sur la configuration de leur distribution Ubuntu, en offrant la possibilité d'ajouter, supprimer ou modifier des logiciels, de paramétrer des options système spécifiques et de mettre en œuvre d'autres ajustements selon leurs besoins spécifiques. Cette personnalisation fine permet de créer des images ISO sur mesure, adaptées aux exigences particulières des utilisateurs, que ce soit pour des projets spécifiques, des déploiements en entreprise ou simplement pour une expérience utilisateur unique.

Cubic se distingue par sa simplicité d'utilisation, rendant le processus de personnalisation accessible même aux utilisateurs moins expérimentés. Son interface graphique intuitive guide les utilisateurs à travers les différentes étapes du processus, facilitant ainsi la création d'une distribution Ubuntu entièrement adaptée à des besoins spécifiques. L'outil prend en charge diverses personnalisations, des ajustements logiciels aux paramètres système avancés, offrant ainsi une flexibilité totale dans la création d'une distribution Ubuntu sur mesure.

En résumé, Cubic représente une solution complète et conviviale pour la création d'images ISO d'Ubuntu personnalisées, mettant entre les mains des utilisateurs un moyen puissant de façonner leur expérience Linux selon leurs préférences et exigences particulières.

2 Personnalisation de la distribution linux avec ubuntu et cubic

Dans le cadre du développement de la distribution personnalisée "ID1FS", le processus a commencé en travaillant sur l'environnement Ubuntu. Nous avons installé l'image ISO d'Ubuntu sur une machine Linux, établissant ainsi la base du système d'exploitation. Pour affiner cette distribution selon les spécifications du projet, nous avons utilisé l'outil Cubic.

Cubic a joué un rôle clé en permettant une personnalisation approfondie de l'image Ubuntu. Cubic a agi comme un atelier de personnalisation, permettant d'ajuster la distribution Linux en fonction des besoins précis du projet. L'outil a facilité la configuration des logiciels préinstallés, la définition des paramètres système et la modification de l'interface utilisateur. Avec Cubic, nous avons pu créer une distribution Ubuntu sur mesure, adaptée aux exigences particulières de "ID1FS". Cette étape de personnalisation a contribué de manière significative à l'élaboration d'une distribution Linux optimisée et unique.

L'ensemble du processus de développement de la distribution personnalisée "ID1FS" implique plusieurs étapes clés. Voici comment ces étapes ont été organisées de manière structurée :

**Étape 1** : Installation de l'image ISO d'Ubuntu sur l'Unix

Nous avons débuté le processus en installant l'image ISO d'Ubuntu sur l'Unix. Cette étape a posé les bases du système d'exploitation sur lequel nous avons construit notre distribution personnalisée.

**Étape 2** : Installation de Cubic

La prochaine étape a été l'installation de Cubic, un outil essentiel pour personnaliser en profondeur l'image d'Ubuntu. Les commandes suivantes ont été utilisées pour installer Cubic :

sudo add-apt-repository ppa:cubic-wizard/release

sudo apt update

sudo apt install cubic

Une fois installé, Cubic a été accessible en tapant simplement "cubic" dans le terminal.

**Étape 3** : Personnalisation de l'image ISO d'Ubuntu avec Cubic

Cubic a joué un rôle central en permettant une personnalisation approfondie de l'image d'Ubuntu. En agissant comme un atelier de personnalisation, Cubic a facilité la configuration des logiciels préinstallés, la définition des paramètres système, et la modification de l'interface utilisateur. Cette personnalisation a été adaptée aux besoins spécifiques de "ID1FS", contribuant significativement à la création d'une distribution Linux optimisée et unique.

**Étape 4** : Transfert de l'image personnalisée vers l'environnement Windows

Une fois la personnalisation terminée, nous avons transféré l'image modifiée depuis l'environnement Linux vers un environnement Windows. Cette démarche a été entreprise pour intégrer l'image personnalisée dans une machine virtuelle sur la plateforme Windows, offrant ainsi un environnement virtuel dédié pour des tests approfondis et une évaluation du système.

**Étape 5** : Installation de l'image personnalisée sur la machine virtuelle

La cinquième étape consistait à installer l'image personnalisée d'Ubuntu, créée avec Cubic, sur une machine virtuelle Windows. Cette opération cruciale permettait d'établir un environnement virtuel dédié pour des tests approfondis et une évaluation complète du système "ID1FS". Le processus impliquait le transfert de l'image modifiée de l'environnement Linux vers la plateforme Windows, suivi des étapes d'importation et de configuration dans la machine virtuelle, assurant une intégration fluide et adaptée aux essais du système personnalisé.

3 Design personnalisé de la distribution Ubuntu ’’ID1FS’’

GNOME, abréviation de GNU Network Object Model Environment, est un environnement de bureau libre et open source qui offre une interface graphique conviviale pour les systèmes d'exploitation basés sur Unix. Il est conçu pour offrir une expérience utilisateur intuitive et flexible, facilitant la navigation et l'interaction avec le système. À la base de GNOME se trouve le gestionnaire de fenêtres GNOME Shell, qui fournit une interface utilisateur moderne et élégante.

Dans le cadre du développement de la distribution personnalisée "ID1FS", nous avons exploité les fonctionnalités de personnalisation de GNOME pour conférer à notre système une identité visuelle unique. En intégrant GNOME Tweaks, un outil puissant d'ajustement des paramètres de l'environnement GNOME, nous avons donné aux utilisateurs de "ID1FS" la possibilité de personnaliser l'apparence de leur bureau selon leurs préférences.

L'un des aspects marquants de cette personnalisation a été l'incorporation du thème "Sweet-Mars" via GNOME Tweaks. Ce thème a été soigneusement choisi pour son esthétique élégante et sa complémentarité avec l'image spécifique "ID1FS" utilisée comme arrière-plan par défaut. L'utilisation transparente de GNOME Tweaks a simplifié le processus d'activation et d'application du thème, offrant ainsi aux utilisateurs une expérience visuelle cohérente et esthétiquement plaisante.

Notre approche a également mis l'accent sur la facilité d'accès à la bibliothèque de thèmes sur le site "GNOME-Look.org" à travers le navigateur Firefox. Cette intégration transparente a permis aux utilisateurs de parcourir et de sélectionner rapidement divers thèmes, renforçant ainsi la personnalisation de leur environnement de bureau. Dans l'ensemble, l'utilisation judicieuse des outils de personnalisation de GNOME a joué un rôle clé dans la création d'une identité visuelle distinctive pour la distribution "ID1FS".

4 Fonctionnement de la distribution

L'intégration et le déploiement du terminal personnalisé dans le cadre du projet "ID1FS" représentent un volet fondamental de la conception axée sur l'expérience utilisateur. Lorsque l'utilisateur lance la commande "terminal.py", le script associé orchestre un processus automatisé qui donne naissance à une fenêtre interactive proposant diverses options. Ces options cruciales comprennent la possibilité de se connecter à un compte existant grâce à l'option "1-log in", de créer un nouveau compte par le biais de l'option "2-registre", ou de simplement quitter le terminal avec l'option "3-exit".

Dans le scénario où l'utilisateur souhaite créer un nouveau compte en choisissant "2-registre", le terminal sollicite la saisie d'un nom d'utilisateur et d'un mot de passe, amorçant ainsi le processus de création de compte. D'autre part, l'option "1-logging" permet à l'utilisateur de se connecter en fournissant ses identifiants existants. Une fois cette étape accomplie, la fenêtre de choix se dissipe, ouvrant la voie à un terminal personnalisé, arborant fièrement le nom d'utilisateur de l'accédant.

À partir de cet instant, l'utilisateur détient les rênes du terminal personnalisé, lui offrant la liberté d'entrer diverses commandes et de façonner son expérience utilisateur selon ses préférences. La personnalisation de cet environnement va au-delà de la simple fonctionnalité, elle incarne un principe fondamental qui place l'utilisateur au cœur de l'interaction avec le système. Cette personnalisation offre une expérience utilisateur fluide, intuitive et adaptable, en totale harmonie avec les exigences spécifiques de "ID1FS". Elle souligne ainsi l'engagement du projet envers une approche conviviale, accessible et sur mesure pour les utilisateurs.