**Université Abdelmalek Essaadi**

**Ecole Nationale des Sciences Appliquées A**



**Première Année Ingénierie des données**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SUJET :**

**Système de fichiers et distribution personnalisée**

**DISTRO-FS**

* **Realisé par :** ASSABBAR MALAK
* **Encadré par :** Pr. Mohamed CHERRADI

**Remerciements**

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet, en particulier à mon professeur, **M. Mohamed CHERRADI**. Ses enseignements et son accompagnement ont été d'une valeur inestimable, m'apportant non seulement des connaissances approfondies en informatique, mais également des compétences avancées en programmation Python et une compréhension approfondie de l'architecture des systèmes de fichiers.

Je souhaite également remercier tous ceux qui ont partagé leurs connaissances et ressources, contribuant ainsi à l'enrichissement de ce travail. Même si j'ai travaillé de manière indépendante, l'inspiration et le support provenant de diverses sources ont été cruciaux pour la réussite de ce projet.

Enfin, nous tenons à remercier Dieu pour nous avoir guidés à travers toutes les difficultés. Nous avons ressenti Sa guidance chaque jour, et c'est grâce à Lui que nous avons pu achever ce parcours académique. Nous plaçons notre confiance en Lui pour le futur qui s'annonce.

**Résumé**

Ce projet avait pour objectif de concevoir et mettre en œuvre un système de fichiers personnalisé, puis de l'intégrer dans une distribution Linux sur mesure. Le projet a impliqué plusieurs étapes clés, notamment la conceptualisation de la structure du système de fichiers, le développement du système de fichiers en utilisant un langage de programmation choisi, les tests de sa fonctionnalité, l'intégration dans une distribution Linux, la création d'une image personnalisée de distribution Linux, l'installation sur une machine virtuelle ou un ordinateur physique, la documentation du processus de développement, et enfin, la présentation et la démonstration du projet.

Le projet a exigé des connaissances avancées en programmation système, en administration Linux, en compilation de noyau, en gestion de paquets, et bien plus encore. Le développement du système de fichiers et de la distribution Linux personnalisée a renforcé les compétences en programmation et approfondi la compréhension de l'architecture informatique.

Ce rapport détaillé du projet relate la planification, l'exécution et les résultats de chaque phase du projet. Il sert de témoignage des compétences acquises, des défis surmontés et de l'intégration réussie d'un système de fichiers personnalisé dans un environnement Linux. Le rapport reconnaît également l'orientation et le soutien reçus du professeur et exprime sa gratitude envers ceux qui ont contribué au succès du projet.

**Table des matière**

**Introduction generale** ............................................................................................................................2

**organisation de rapport**:.........................................................................................................................3

**Chapitre 1 :système de fichiers** .............................................................................................................5

* Diagramme de cas d’utilisation :……………………………………………………………………………………… 6
* Technologie…………………………………………………………………………………………………………………….…8
* Herarchie……………………………………………………………………………………………………………………….…11
* Login……………………………………………………………………………………………………………………………… 13
* Login………………………………………………………………………………………………………………………..………14
* Commande………………………………………………………………………………………………………………………15
* Conclusion……………………………………………………………………………………………………………………….20

**Chapitre 2 : distribution :** ………………………………………………………………………………………………………………. 22

* Choix d’outils : …………………………………………………………………………………………………………….….25
* Personnalisation avec cubic…………………………………………………………………………………………….26
* Definition de cubic………………………………………………………………………………………………………..…28
* instalation………………………………………………………………………………………………………………………..29
* personnalisation du theme……………………………………………………………………………………………..31

**Introduction générale**

En tant qu'étudiant en première année de spécialisation en genie Transformation Digitale et Intelligence Artificielle, j'ai entrepris un projet ambitieux intitulé **"DISTRO-FS"** visant à créer un système de fichiers personnalisé et à l'intégrer dans une distribution Linux sur une machine virtuelle. Ce projet m'a offert l'opportunité d'explorer divers aspects de la programmation système, de l'administration Linux, de la compilation de noyau, de la gestion de paquets, et bien plus encore.

La conception du système de fichiers a constitué une étape cruciale, nécessitant une réflexion approfondie sur la structure, les fonctionnalités, la hiérarchie des répertoires, les permissions, et les types de fichiers pris en charge. Le développement du système de fichiers, réalisé en utilisant le langage de programmation Python, a été suivi de tests approfondis pour garantir son bon fonctionnement.

La création d'une image de distribution (.iso) a permis d'aboutir à une solution autonome, prête à être installée sur une machine virtuelle ou un ordinateur physique.

La documentation méticuleuse du processus de développement, d'intégration et d'installation, ainsi que la réalisation de présentations et démonstrations, ont été des étapes essentielles pour partager mon travail avec d'autres et mettre en lumière les fonctionnalités du système de fichiers personnalisé.

Ce projet a été une expérience enrichissante, combinant des connaissances théoriques avec des compétences pratiques, et représentant un jalon significatif dans mon parcours académique en génie en Transformation Digitale et Intelligence Artificielle .

**ORGANISATION DU RAPPORT**

Ce rapport est une synthèse des travaux réalisés au cours du projet. Il est organisé en deux chapitres :

**• Premier chapitre :** **- Système de Fichiers Personnalisé (TDIA-FS) :**

Ce chapitre expose la conception, le développement, les tests et les fonctionnalités du système de fichiers personnalisé TDIA-FS, mettant en avant sa structure, son fonctionnement, et les choix architecturaux opérés.

**• Deuxième chapitre :** **- Distribution Linux Personnalisée (TDIA-DISTRO) :**

Le deuxième chapitre se focalise sur la création et l'intégration d'une distribution Linux personnalisée, détaillant les étapes de développement , de création d'une image ISO, et d'installation sur une machine virtuelle ou physique.

**Chapitre I : SYSTÈME DE FICHIERS:**

* **Idée : Liberté de Navigation :**

**Dans le cadre novateur de ce projet, l'idée maîtresse résidait dans la conception d'un système de fichiers offrant une navigation intuitive et sans contrainte, éliminant ainsi le besoin fastidieux de spécifier des chemins d'accès pour chaque action. Contrairement aux systèmes conventionnels où la manipulation de fichiers et de répertoires exige la précision d'un chemin, notre approche révolutionnaire permet à l'utilisateur de naviguer librement dans l'ensemble du système en se concentrant simplement sur le nom du répertoire ou du fichier ciblé. Cette conception innovante vise à simplifier l'expérience de l'utilisateur, offrant une flexibilité inédite et éliminant les contraintes liées à la gestion classique des répertoires. Ce concept novateur représente un saut qualitatif dans la convivialité et la praticité des opérations effectuées dans le cadre du système de fichiers personnalisé.**

**SYSTÈME DE FICHIERS**

* **SYSTEME DE FICHIER :**

Lors d'une migration d'un autre système d'exploitation tel que Microsoft Windows vers un autre, une chose qui impactera considérablement l'utilisateur final sera les différences entre les systèmes de fichiers.

C’EST QUOI UN SYSTÈME DE FICHIERS ?

Le système de fichiers constitue l'épine dorsale de la gestion des données dans un système informatique. Il agit comme une infrastructure organisée qui permet le stockage, l'organisation et l'accès aux fichiers et répertoires. À travers une hiérarchie structurée, le système de fichiers offre aux utilisateurs et aux applications un moyen intuitif d'interagir avec les données stockées. Il englobe des concepts tels que la création, la lecture, l'écriture, et la suppression de fichiers, ainsi que la gestion des permissions et des métadonnées.

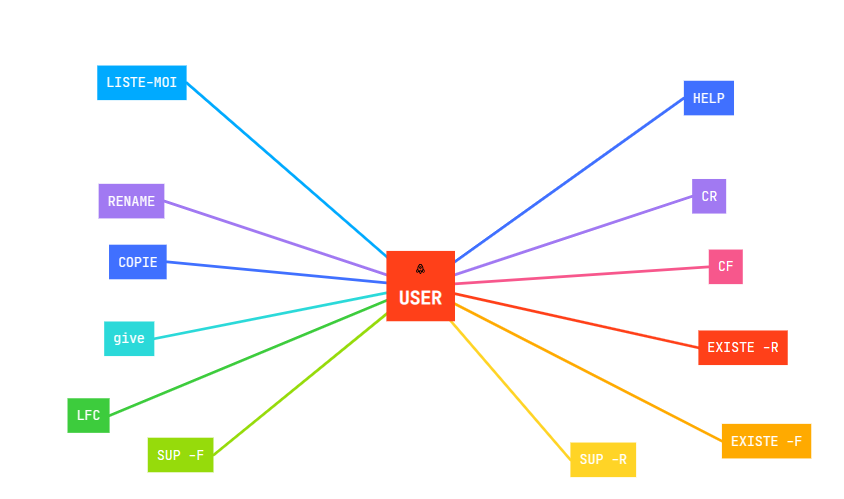
C’EST QUOI LE ROLE D’UN SYSTÈME DE FICHIERS ?

Le rôle fondamental du système de fichiers est de fournir une méthode cohérente et efficace pour stocker, récupérer et organiser des données sur un support de stockage, qu'il s'agisse de disques durs, de SSD ou d'autres dispositifs de stockage. Il permet aux utilisateurs de structurer leurs informations de manière logique, facilitant ainsi la gestion des fichiers et répertoires. En agissant comme une interface entre l'utilisateur et le matériel de stockage, le système de fichiers simplifie les opérations courantes tout en garantissant l'intégrité et la sécurité des données .

**SYSTÈME DE FICHIERS**

­­­­­­­­­­­­­­­­­­

1. **DIAGRAMME DE CAS D’UTILISATION :**

Dans cette partie du diagramme d'utilisation, l'utilisateur interagit de manière transparente avec le système de fichiers personnalisé en effectuant diverses actions, telles que la création de répertoires, la gestion de fichiers, la vérification d'existence, la modification des droits, et bien d'autres. Cette représentation visuelle met en lumière la convivialité et la diversité des fonctionnalités offertes par le système, offrant à l'utilisateur un contrôle complet sur ses données et ses opérations.

* **HELP :** affiche un message d'aide dans la console, listant les commandes possibles

dans le programme et comment elles sont utilisées .

* **CR :** créer un nouveau répertoire dans le système de fichiers en spécifiant un nom.
* **CF :** créer un nouveau fichier dans le système de fichiers en spécifiant un nom.
* **COPIE**: L'utilisateur peut copier un fichier ou un répertoire .
* **SUP**: L'utilisateur peut supprimer un répertoire (SUP -R) ou un fichier (SUP -F)

existant dans le système de fichiers.

* **EXISTE :** chercher l’existence d’un fichier (EXISTE -F) ou un repertoire (EXISTE -R) dans

système de fichiers .

* **RENAME** : Cette fonctionnalité permet à l'utilisateur de renommer un répertoire ou

un fichier.

* **GIVE :** L'utilisateur peut attribuer des droits spécifiques (lecture (+r), écriture(+w),

exécution (+e)) à un fichier.

* **LFC :** L'utilisateur peut afficher le contenu d'un fichier .

**SYSTÈME DE FICHIERS**

1. **TECHNOLOGIES :**

* **Pourquoi LINUX ?**

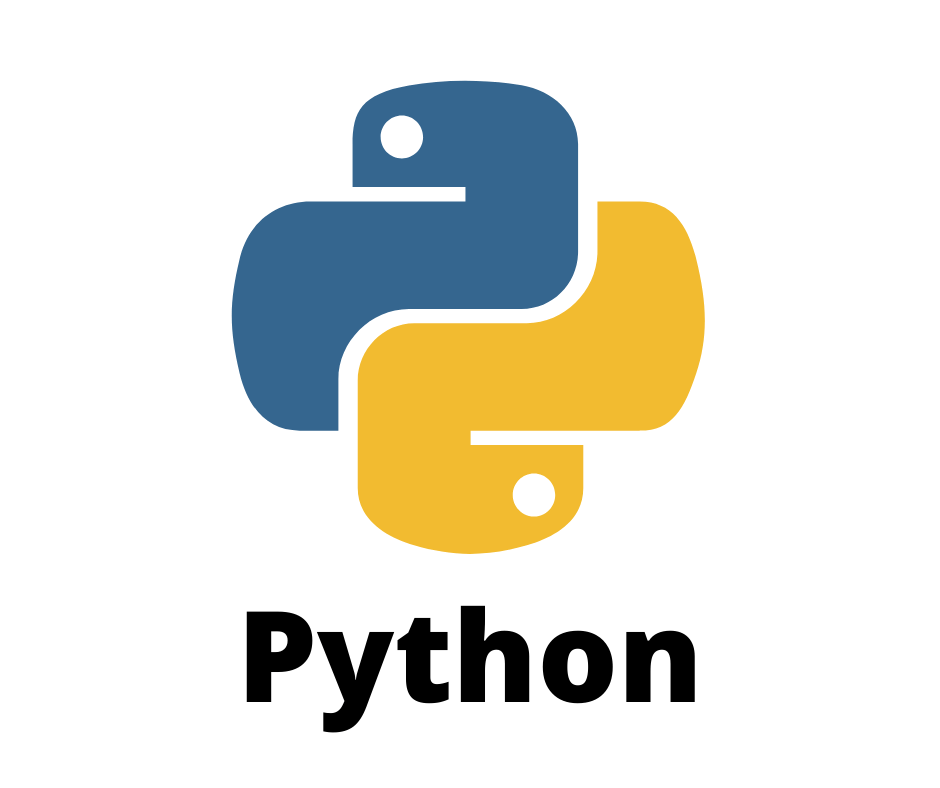
Le développement de notre système de fichiers personnalisé a été réalisé sous l'environnement Linux, une plateforme robuste et open-source largement utilisée dans le domaine de l'informatique.

* + - **CHOIX DE LINUX ?**

**Linux**, un système d'exploitation open-source, est largement reconnu pour sa stabilité, sa sécurité et sa flexibilité. En tant qu'alternative populaire à d'autres systèmes d'exploitation, Linux offre un environnement de développement puissant et personnalisable, idéal pour la création de projets complexes. Son modèle open-source permet un accès étendu aux ressources et outils, favorisant l'innovation et la collaboration communautaire. Avec une compatibilité étendue, Linux offre une solution polyvalente pour diverses architectures matérielles, offrant ainsi une plateforme solide pour le développement de projets informatiques.

**SYSTÈME DE FICHIERS**

* **Pourquoi python ?**

**Python,** un langage de programmation interprété, polyvalent et convivial, est largement utilisé dans divers domaines tels que le développement web, l'automatisation, la science des données et la programmation système. Sa syntaxe claire et concise, ainsi que sa vaste bibliothèque standard, en font un choix attrayant pour les développeurs.

* + - * **CHOIX DE PYTHON ?**

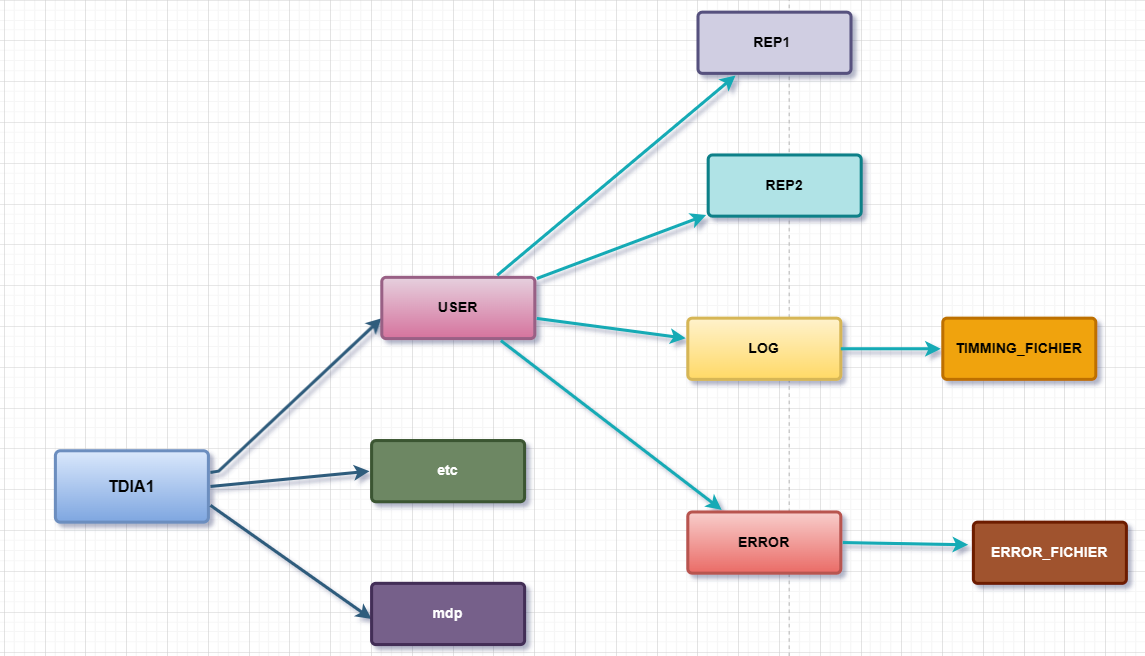
Le choix de Python pour ce projet repose sur plusieurs considérations :

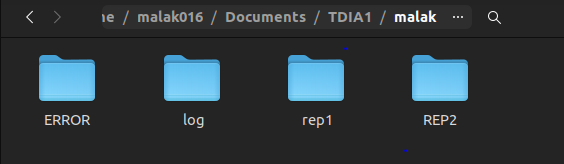
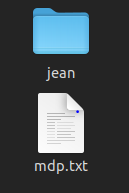
* la simplicité de la syntaxe de Python facilite le développement rapide et la compréhension du code, ce qui est crucial pour un projet de cette envergure.
* la grande communauté Python et son écosystème étendu de bibliothèques offrent un support et des outils significatifs.
* La portabilité de Python est également un avantage, permettant l'exécution du code sur différentes plateformes sans modification significative .
* Sa flexibilité en termes de paradigmes de programmation (orienté objet, impératif, fonctionnel) offre une approche adaptable aux besoins spécifiques du projet.
* En outre, Python dispose de bibliothèques puissantes pour la manipulation de fichiers, la gestion de systèmes de fichiers, et la mise en réseau, ce qui est essentiel pour le développement d'un système de fichiers personnalisé et la création d'une distribution Linux.

En résumé, le choix de **Python** s'appuie sur sa simplicité, sa portabilité, sa communauté active et son ensemble d'outils, faisant de lui un langage idéal pour concevoir et mettre en œuvre un projet aussi complexe.

**SYSTÈME DE FICHIERS**

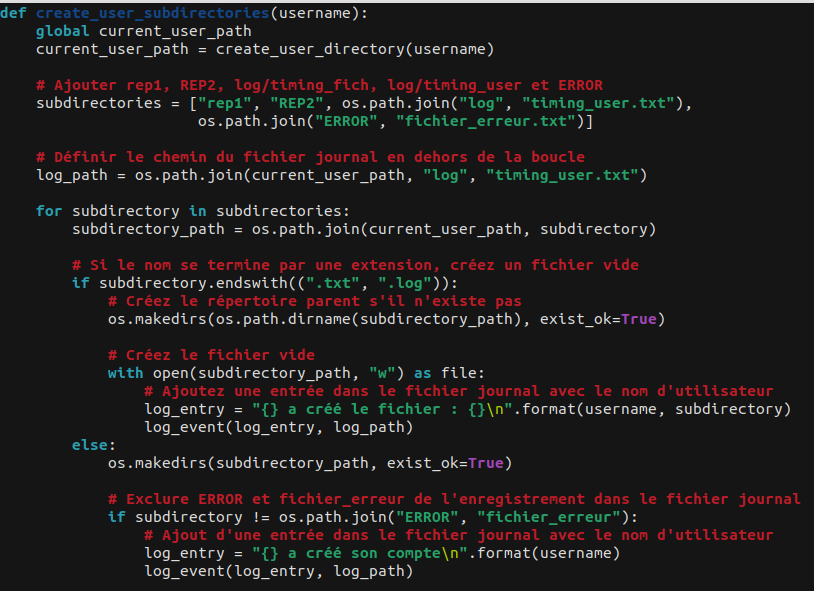
1. **HIÉRARCHIE DU SYSTÈME DE FICHIERS :**

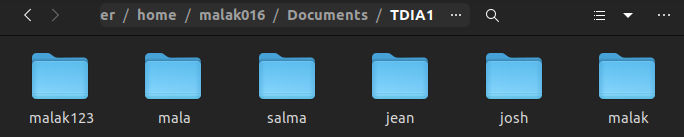
L'architecture du système de fichiers, telle une toile ingénieusement tissée, définit la structure fondamentale de notre projet **"TDIA-FS".** Plongeons dans cette hiérarchie savamment orchestrée, où chaque répertoire et fichier contribue à l'équilibre dynamique de notre système, créant ainsi une toile numérique propice à l'organisation et à l'efficacité des opérations. Explorez avec moi cette structure ingénieuse qui sous-tend notre système de fichiers personnalisé, sculptant un paysage virtuel où chaque composant a son rôle et sa place bien définie.

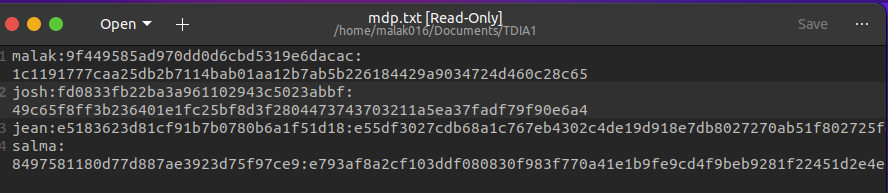


La conception de la hiérarchie du système de fichiers a été réalisée avec soin, en définissant une structure qui favorise la clarté et l'efficacité. L'implémentation a été effectuée en utilisant le langage de programmation Python, où chaque répertoire et fichier a été créé avec une attention particulière aux détails fonctionnels et organisationnels. Le code correspondant .

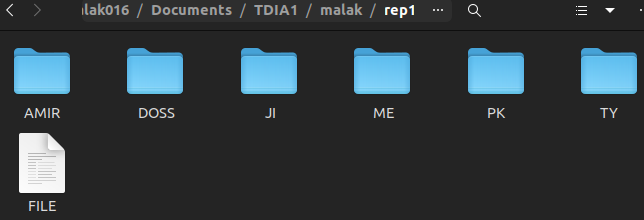
Dans le but de mettre en place cette structure de répertoire, nous avons rédigé le code Python suivant en portant une attention méticuleuse aux détails fonctionnels et organisationnels :



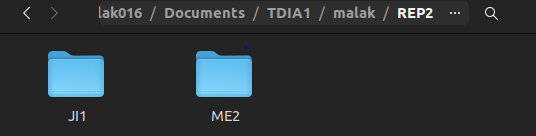
* **USER :** Le répertoire utilisateur (user directory) est un répertoire personnel attribué à chaque utilisateur après la création de son compte donc ce repertoire est conçu pour fournir à chaque utilisateur un espace dédié où il peut organiser et stocker ses propres fichiers et dossiers de manière sécurisée. Lorsqu'un utilisateur crée un compte avec un nom d'utilisateur spécifique, un répertoire portant ce nom est automatiquement créé dans le répertoire utilisateur.
* **Mdp :** un fichier qui contient le mot de passe hashé de tous les utilisateurs existe dans le système fichiers .

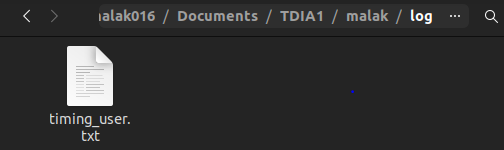


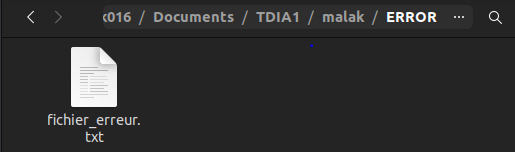
* **REP1 :** Contient les répertoires que chaque utilisateur crée dans votre système donc Les utilisateurs peuvent organiser leurs fichiers et dossiers dans ce répertoire.



* **REP2 :** Stocke des copies de fichiers et de répertoires ainsi que les sous repertoires que l'utilisateur souhaite sauvegarder. Offre aux utilisateurs un moyen de sauvegarder leurs données importantes.



* **LOG :** Ce répertoire est spécialisé dans la gestion des événements système. Enregistre les actions de l'utilisateur, telles que la création ou l'ouverture de fichiers. Les informations détaillées comme l'heure et le nom d'utilisateur sont stockées. Et aussi le chngement des permissions .
* **ERROR :** Enregistre les erreurs commises par l'utilisateur lors du démarrage du système . Cela peut inclure des commandes mal écrites ou des erreurs d'utilisation.



**SYSTÈME DE FICHIERS**

1. **Login :**

La partie "**login**" de notre système de fichiers a été méticuleusement élaborée pour offrir aux utilisateurs une expérience d'authentification sécurisée et personnalisée. Voici un aperçu des caractéristiques principales :

L'authentification est assurée par l'utilisation de noms d'utilisateur et de mots de passe standard, avec la possibilité d'opter pour une authentification à deux facteurs pour une sécurité renforcée. La personnalisation de l'apparence de l'interface de connexion a été introduite, offrant aux utilisateurs la liberté de choisir un thème qui reflète leurs préférences, tel qu'un thème inspiré de macOS. Ainsi que Des messages d'erreur clairs et intuitifs guident les utilisateurs en cas d'informations d'identification incorrectes, assurant une résolution rapide des problèmes. L'interface de connexion est conçue pour une compatibilité multiplateforme, garantissant une expérience utilisateur homogène sur divers dispositifs et navigateurs. Les utilisateurs peuvent personnaliser l'arrière-plan de leur interface de connexion, que ce soit en choisissant parmi des images prédéfinies ou en important leurs propres images.

Chaque tentative de connexion est minutieusement consignée, enregistrant des détails tels que l'heure, le nom d'utilisateur, et les tentatives infructueuses. Ces journaux sont mis à la disposition des administrateurs pour des vérifications de sécurité.

Cette approche vise à offrir une interface de connexion qui harmonise sécurité, convivialité, et personnalisation, contribuant ainsi à une expérience d'authentification fiable et agréable pour nos utilisateurs.



1. **mzmzmzdbhjusgduygsnbdhgyj**

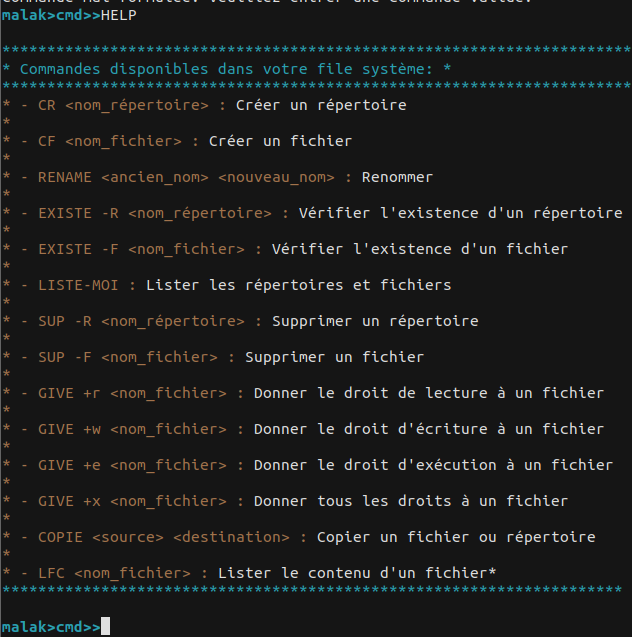
**SYSTÈME DE FICHIERS**

1. **LES COMMANDES :**

La section des commandes représente l'interface essentielle permettant aux utilisateurs d'interagir de manière dynamique avec le système de fichiers personnalisé. À travers cette interface en ligne de commande, les utilisateurs peuvent exécuter diverses opérations telles que la création de répertoires, la gestion de fichiers, la navigation dans la structure du système, et bien plus encore. Cette partie joue un rôle central dans l'expérience utilisateur, offrant un moyen puissant et flexible de manipuler le système.

Voici les principales commandes que nous avons créées pour la navigation :

**HELP :** **Afficher l’utilisation des commandes .**

La commande ' affiche les informations et les instructions disponibles pour vous guider dans l'utilisation du système.

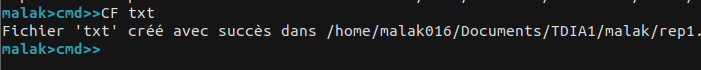
**CR : créer un repertoire**

C’est pour créer un nouveau répertoire dans le répertoire actuel. Cela vous permet d'organiser nos fichiers de manière logique et de créer une structure hiérarchique.



**CF : créer un fichier**

En utilisant la commande "CF", vous créez un fichier vide . C'est utile lorsque vous avez besoin d'un espace pour stocker des données spécifiques .



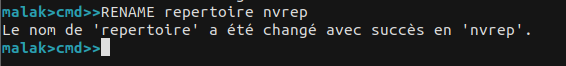
**RENAME : renomer un repertoire ou fichier**

Cette commande nous permet de renommer un répertoire ou un fichier. Cela peut

être utile lorsque vous avez besoin de mettre à jour les noms pour mieux refléter le

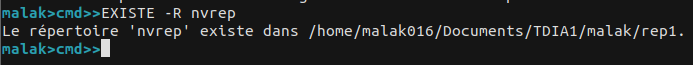
contenu.





**EXISTE -R : Rechercher un repertoire**

En utilisant cette commande , vous pouvez vérifier si un répertoire spécifié existe dans le répertoire actuel. Cela vous aide à confirmer la présence d'une structure de répertoires particulière.



**EXISTE -F : Rechercher un fichier**

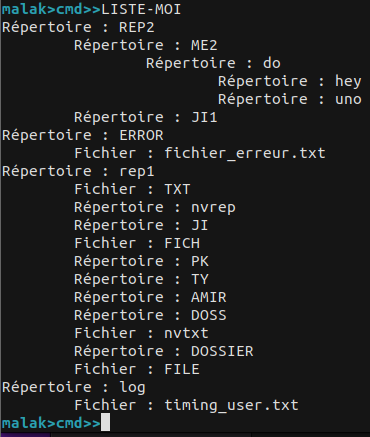
vous pouvez vérifier si un fichier spécifié existe dans le répertoire actuel. Cela vous donne la possibilité de confirmer la présence d'un fichier particulier.

Définition de la commande : "EXISTE" signifie "Existe", et "-F" indique que la recherche concerne les fichiers.



**LISTE-MOI : Lister les reprtoires et les fichiers .**

Lorsque vous utilisez "LISTE-MOI", le système affiche la liste complète des répertoires et fichiers présents dans système de fichiers . Cela offre une vue d'ensemble de la structure actuelle.



**SUP -R : Supprimer un Reprtoire .**

Cette commande vous permet de supprimer un répertoire spécifié ainsi que tous ses sous-répertoires et fichiers il suffit de presiser que le nom de repertoire . C'est une action irréversible, alors utilisez-la avec précaution.



**SUP -F : Supprimer un Fichier .**

En utilisant "SUP -F", vous supprimez un fichier spécifié Comme pour la suppression de répertoire, soyez prudent car c'est une action irréversible.



**GIVE +r : donner la permission de lire .**

La commande ajoute le droit de lecture au fichier spécifié. Cela permet aux utilisateurs d'accéder au contenu du fichier sans pouvoir le modifier.



**GIVE +w : donner la permission d’ecrire .**

vous ajoutez le droit d'écriture au fichier spécifié. Cela autorise la modification du contenu du fichier.



**GIVE +e : donner la permission d'exécution .**

La commande ajoute le droit d'exécution au fichier spécifié. Cela permet à l'utilisateur d'exécuter le fichier s'il s'agit d'un script ou d'un exécutable.

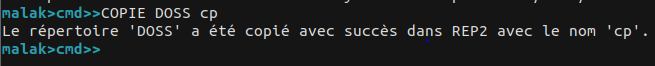


**GIVE +x : donner tous les permissions .**

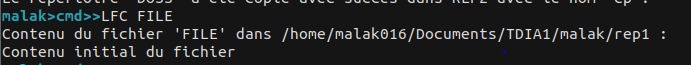
 vous ajoutez tous les droits au fichier spécifié, permettant à l'utilisateur de lire, écrire et exécuter le fichier.

**COPIE :** **copie repertoire ou fichier .**

La commande "COPIE" vous permet de copier un fichier ou un répertoire d'un emplacement à un autre, facilitant la duplication du contenu.



**LFC : lister le contenu d’un fichier .**

 En utilisant "LFC", vous affichez le contenu d'un fichier directement dans la console, offrant une vue rapide des données.

**EXIT : QUITTER .**



1. **CONCLUSION :**

La conception et l'implémentation du système de fichiers "TDIA1" reflètent notre engagement envers une gestion efficace et sécurisée des données. En adoptant une architecture organisée avec des répertoires bien définis, nous avons créé un environnement où la structure répond aux besoins fonctionnels tout en garantissant une expérience utilisateur intuitive.

La hiérarchie, avec des répertoires tels que "Accounts", "REP1", "REP2", "COUPLE", et "ERROR", offre une gestion claire des comptes utilisateurs, des répertoires personnels, des sauvegardes, des événements système, et des erreurs. Chacun de ces composants contribue à la cohérence et à l'efficacité de notre système.

L'aspect personnalisable du système se manifeste également dans l'interface de connexion, où la personnalisation est intégrée pour offrir une expérience d'authentification non seulement sécurisée, mais aussi esthétiquement adaptée aux préférences des utilisateurs.

En fin de compte, le système de fichiers "TDIA1-FS" représente une fusion réussie entre organisation structurée et adaptabilité, visant à répondre aux exigences fonctionnelles tout en offrant une expérience utilisateur agréable. Cette approche équilibrée souligne notre engagement envers l'efficacité, la sécurité et la satisfaction globale des utilisateurs.

**Chapitre 2 : DISTRIBUTION :**

La création d'une distribution Linux personnalisée représente une étape cruciale dans notre projet, apportant une dimension unique et adaptée à nos besoins spécifiques. Cette section met en lumière les choix stratégiques, les outils de personnalisation, et les caractéristiques clés qui ont guidé la création de notre distribution basée sur Ubuntu. En détaillant les étapes de personnalisation, les ajustements apportés à l'environnement de bureau, et la sélection minutieuse des applications préinstallées, nous démontrons l'engagement envers la création d'une expérience utilisateur sur mesure. Cette distribution personnalisée, façonnée par des choix réfléchis et des considérations fonctionnelles, incarne notre vision de fournir un système d'exploitation qui répond spécifiquement aux besoins de notre communauté d'utilisateurs.

**DISTRIBUTION**

1. **CHOIX D’OUTILS :**

* Pourquoi ubuntu ?

Le choix de la distribution Ubuntu comme fondation pour notre distribution personnalisée découle de ses avantages significatifs, notamment sa stabilité reconnue, sa vaste communauté active, son écosystème logiciel étendu, sa flexibilité de personnalisation, sa documentation abondante et sa compatibilité matérielle diversifiée. Ces caractéristiques assurent une base solide et adaptable pour notre projet, permettant une création efficace et une expérience utilisateur optimale.

* Pourquoi cubic ?

Cubic a été choisi en raison de sa facilité d'utilisation, de son interface graphique intuitive, de sa capacité à générer des images ISO et de sa compatibilité directe avec Ubuntu. Ces aspects ont grandement simplifié le processus de personnalisation de notre distribution, contribuant ainsi à une mise en œuvre efficace de notre vision projet.

**DISTRIBUTION**

**2 – PERSONNALISATION AVEC CUBIC :**

* Definition de cubic :

Cubic est un utilitaire avancé de personnalisation de distributions Linux, spécifiquement conçu pour la plateforme Ubuntu. Cet outil offre une interface graphique conviviale, permettant aux utilisateurs de façonner et d'ajuster divers aspects de leur distribution personnalisée.

Le processus de personnalisation avec Cubic englobe la sélection fine des composants du système, la configuration des paramètres de l'environnement de bureau, l'intégration d'applications spécifiques, et la mise en place de divers scripts personnalisés. Cubic simplifie ce processus en offrant une navigation visuelle intuitive, réduisant la dépendance aux commandes en ligne et facilitant l'accès aux utilisateurs de différents niveaux d'expertise.

L'une des fonctionnalités clés de Cubic réside dans sa capacité à générer des images ISO. Cette fonctionnalité est particulièrement utile car elle permet aux utilisateurs de créer des versions autonomes de leur distribution personnalisée, prêtes à être installées sur une variété de systèmes sans nécessiter des étapes de configuration complexes.

En résumé, Cubic se positionne comme un outil puissant, offrant une approche graphique et conviviale pour la personnalisation de distributions Linux basées sur Ubuntu, facilitant ainsi la création d'environnements sur mesure adaptés aux besoins spécifiques des utilisateurs.

* Installation et Personnalisation avec Cubic sur Ubuntu :

nous plongeons dans les coulisses de la création sur mesure de notre distribution Linux. En se concentrant sur l'utilisation de l'outil puissant qu'est Cubic sur la plateforme Ubuntu, cette partie détaille les étapes d'installation et de personnalisation qui ont été le catalyseur de notre distribution unique. De la configuration initiale à la génération d'une image ISO prête à être déployée, suivez le parcours captivant de la transformation d'une distribution de base en une expérience utilisateur sur mesure. Ces étapes, soigneusement orchestrées avec Cubic, incarnent l'essence même de notre engagement envers l'innovation et la création d'un système d'exploitation adapté à nos besoins spécifiques.

Etape 1 :

J’Ouvre le terminal sur système Ubuntu.On Utilisons la commande sudo apt update pour mettre à jour la liste des paquets. Ensuite, installez Cubic en utilisant la commande sudo apt install cubic.

Etape 2 : lancement cubic

Une fois l'installation terminée, j’ai lance utilisant la commande cubic dans le terminal ou en le recherchant dans le menu des applications.

Etape 3 : selection de l’image de base

Au lancement de Cubic, je suis invité à sélectionner une image de base. J’ai choisi une image Ubuntu qui correspond à la version sur laquelle vous souhaitez baser sur ma distribution personnalisée.

Etape 4 : Configuration de l’environnement de base

Cubic vous permettra ensuite de configurer l'environnement chroot, ce qui est essentiel pour effectuer des modifications dans l'image de base. Suivez les instructions à l'écran pour configurer cet environnement .

Etape 5 : personnalisation de l’image

Vous serez ensuite placé dans un environnement chroot, où vous pourrez effectuer des modifications personnalisées. Cela peut inclure l'ajout ou la suppression d'applications, la configuration de l'environnement de bureau, le réglage des paramètres système, etc.

Etape 6 : Configuration de l’image ISO

Une fois que j’ai terminé la personnalisation, j’ai returne la fenêtre principale de Cubic. Vous aurez l'option de générer une nouvelle image ISO. Suivez les instructions pour créer l'image ISO personnalisée.

Etape 7 : Sauvgarde et distrubution

Cubic m’a permet de sauvegarder mon projet, y compris tous les scripts personnalisés et les configurations que j’ai ajoutés. On preut ensuite distribuer cette image ISO ou la déployer sur

d'autres systèmes.

**DISTRIBUTION**

**3 - Personnalisation Esthétique du Thème sous GNOME :**

Au-delà des fonctionnalités essentielles, l'aspect visuel d'une distribution Linux joue un rôle essentiel dans l'expérience utilisateur. Cette section explore le volet esthétique de notre distribution personnalisée, mettant en lumière le processus de changement de thème au sein de l'environnement de bureau GNOME.

En modifiant le thème, nous avons cherché à infuser une identité visuelle distinctive à notre distribution, s'alignant ainsi sur l'esthétique Cubic. Du choix minutieux des couleurs à la sélection des icônes et des polices, chaque ajustement a été soigneusement orchestré pour créer une expérience utilisateur cohérente et agréable.

Cette démarche va au-delà de l'aspect purement fonctionnel pour offrir aux utilisateurs une esthétique visuelle engageante et personnalisée. Suivez le récit de cette transformation visuelle, démontrant comment chaque détail contribue à façonner une distribution Linux qui transcende les attentes, offrant une expérience immersive à chaque interaction.

**Accès aux Paramètres GNOME :**

À l'intérieur de l'environnement chroot, nous avons ouvert les Paramètres GNOME, l'interface utilisateur graphique pour la configuration du bureau.

**Navigation vers l'Outil de Personnalisation :**

Dans les Paramètres GNOME, nous avons localisé l'outil de personnalisation ou l'option dédiée aux thèmes. Cela peut varier selon les versions de GNOME, mais généralement, il peut être trouvé dans la section "Apparence" ou "Personnalisation".

**Sélection du Thème Cubic :**

Nous avons sélectionné un thème spécifique, soit en utilisant un thème préinstallé qui correspond à l'esthétique Cubic, soit en téléchargeant et installant un thème Cubic personnalisé depuis les sources disponibles.

**Configuration des Paramètres de Thème :**

À l'intérieur de l'outil de personnalisation, nous avons configuré les paramètres du thème tels que les couleurs, les polices, les icônes et autres éléments visuels pour créer une esthétique Cubic distinctive.

**Aperçu et Validation :**

Nous avons utilisé la fonction d'aperçu pour visualiser les modifications en temps réel, ajustant les paramètres au besoin pour assurer une harmonie visuelle avec notre vision de distribution.

**Sauvegarde des Configurations :**

Après avoir finalisé les modifications, nous avons enregistré les configurations, assurant que le thème Cubic personnalisé serait appliqué de manière persistante.

Cette approche, spécifique à l'environnement GNOME, a permis d'apporter des changements esthétiques cohérents avec l'esthétique Cubic à notre distribution personnalisée.

**CONCLUSION :**

À travers ce projet captivant, qui a exploré les méandres du système de fichiers et a donné naissance à une distribution Linux personnalisée, un aspect distinctif mérite une attention particulière. La personnalisation effectuée avec l'outil Cubic sur la base d'Ubuntu va au-delà des simples ajustements fonctionnels. En effet, la refonte complète du thème pour adopter un style Cubic personnalisé a été un axe de transformation esthétique significatif.

Cette démarche a ajouté une couche supplémentaire à notre distribution, la rendant non seulement fonctionnelle mais également esthétiquement distinctive. Le changement de thème vers un style Cubic spécifique souligne notre souci du détail et notre engagement à offrir une expérience utilisateur complète. Chaque élément visuel, des icônes au fond d'écran, reflète notre identité et renforce l'esthétique cohérente de notre distribution.

En résumé, cette personnalisation du thème avec Cubic transcende l'aspect purement fonctionnel pour embrasser une approche holistique de la création d'un environnement informatique unique. Ce projet, alliant innovation technique et esthétique personnalisée, renforce notre vision d'un système d'exploitation qui va au-delà des attentes, offrant une expérience utilisateur immersive et distinctive.