

Rapport Projet 2CS

Réalisation d'une distribution Linux réalisant la
fonction principale de terminal intelligent dédiée
pour l'enseignement

Réalisé Par l'équipe 12 :

- HENNOUNI Narimane (CE)
- EL HASSANE Nour
- RAIHAH Amine
- AZIZ Rezak
- SEHDI Nassim

Encadré Par :

1. AMROUCHE Hakim

Table des matières

Table des figures	3
Liste des tableaux	3
Introduction	4
1 Étude Bibliographique	5
1.1 Concepts de base	5
1.2 Approches pour personnaliser Linux	6
1.3 Étude de cas Linux	7
1.3.1 Pourquoi Linux	7
1.3.2 Structure de Linux	7
1.3.3 Architectures Matérielle supportés par Linux	9
1.4 Etude des processeurs ARM	10
1.4.1 La relation entre le système d'exploitation et l'architecture du processeur	10
1.4.2 Architecture de processeur ARM	10
1.4.3 Système embarqué pour ARM	11
1.4.4 Raspberry Pi	11
1.4.5 Contexte d'utilisation de l'architecture ARM	11
1.5 Destributions Linux orientées enseignement	12
2 Analyse des besoins	15
2.1 Besoins techniques :	15
2.2 Besoins fonctionnels :	15
2.3 Besoins utilisateurs :	15
3 Réponses aux besoins	16
3.1 Étude comparative des outils existants	16
3.1.1 Live Build :	16
3.1.2 Linux Live Kit :	17
3.1.3 Pinguy builder :	17

3.1.4	Slax :	18
3.1.5	Ubuntu Imager :	19
3.1.6	Récapitulatif	20
3.2	Étude comparative des applications :	21
3.2.1	Environnement de bureaux :	21
3.2.2	Accès à distance :	22
3.2.3	Serveur vidéo et client :	23
3.2.4	Logiciels bureautiques :	23
3.2.5	Navigateur Web :	24
3.3	Applications utilisateur	24
4	Gestion Projet	26
	Conclusion	29

Table des figures

1.1	Architecture de Linux	7
1.2	Architecture du processeur ARM	10
1.3	Quelques contextes d'utilisation de l'ARM	12
3.1	Logo de Pinguy Builder	18
3.2	Logo de Slax	19
3.3	Logo de Ubuntu Imager	19
4.1	Logo Lykeio	26
4.2	Planning du travail	28

Liste des tableaux

3.1	Avantages et inconvénients de Live Build	17
3.2	Avantages et inconvénients de Linux Live Kit	17
3.3	Avantages et inconvénients de Pinguy builder	18
3.4	Avantages et inconvénient de Slax	19
3.5	Avantages et inconvénients de Ubuntu	20
3.6	Avantages et inconvénients des différents environnement de bureaux	21
3.7	Avantages et inconvénients des différents outils d'accès à distance	22
3.8	Avantages et inconvénients des outils vidéos	23
3.9	Avantages et inconvénients des logiciels bureautiques	23
3.10	Avantages et inconvénients des Navigateurs Web	24

Introduction

Face à la forte croissance du monde de l'informatique, les exigences techniques, économiques, politiques et sociales on se retrouve face à plusieurs problématiques nécessitant des réponses rapides et efficaces mais aussi économiques, sûres et évolutives. La rapidité avec laquelle la technologie évolue dépasse largement la capacité des utilisateurs à l'acquérir. Les entreprises des différents secteurs se retrouvent face à l'obligation de renouveler leurs parcs informatiques, cependant le renouvellement de ce dernier coûte cher surtout dans des milieux ayant peu de ressources financières. De plus, il faut trouver un moyen de se débarrasser des anciennes machines. Plusieurs contraintes s'imposent de différents types : économiques, écologiques ainsi que sécuritaires.

Les écoles étant un milieu dans lequel l'outil informatique est devenu une nécessité, on s'intéresse à exploiter les anciennes machines pour profiter des nouvelles applications ayant des besoins bien exigeants. Une solution proposée est l'utilisation d'une architecture à base de terminaux. Cette solution n'est pas nouvelle mais elle a été beaucoup utilisée au début de l'âge des ordinateurs, ensuite elle a été abandonnée mais la nécessité d'une solution face aux contraintes déjà citées a fait que les entreprises s'intéressent de plus en plus vers ce genre de solution.

On se retrouve donc face à un nouveau problème, la majorité des systèmes existants consomment énormément de ressources. Heureusement la solution pour ce problème n'était pas loin des yeux. Linux, grâce à ses caractéristiques d'open source et sa structure modulaire, permet de créer et d'adapter une distribution selon les besoins bien précis.

Notre étude se focalise sur un niveau précis qui est le niveau secondaire (les lycées). Le but est de créer une distribution linux légère adaptée aux besoins des lycéens et assurant l'exploitation des anciens parc informatiques existant. Pour illustrer notre étude et la démarche suivi nous avons choisi d'organiser la suite de la façon suivante : Nous allons commencer par une étude bibliographique et une étude de l'existant, ensuite nous allons recenser les besoins dont la distribution réalisée doit respecter ainsi que les moyens pour les satisfaire à travers des choix techniques et on finira par une illustration de la gestion de ce projet.

Chapitre 1

Étude Bibliographique

1.1 Concepts de base

1. Système d'exploitation :

Un système d'exploitation est le logiciel principal qui gère tout le matériel et les autres logiciels d'un ordinateur. Il assure le bon fonctionnement de l'appareil puisqu'il joue le rôle d'intermédiaire entre les composants matériels et logiciels, fournit tous les services nécessaires utilisés par les applications et contrôle tout conflit existant entre les programmes.

2. Linux :

Linux est un système d'exploitation créé en 1991 par Linus Torvalds dans le but de créer un clone d'UNIX sans recourir au code de ce dernier. C'est un système d'exploitation préemptif et multitâche, ce qui signifie que vous pouvez avoir plus d'un programme fonctionnant simultanément. Linux se charge de basculer entre les différents processus. Le secret de sa réputation réside dans son code source libre et gratuit ainsi que la possibilité d'en créer de nouvelles versions personnalisées.

3. Système embarqué :

C'est une combinaison de matériels et logiciels permettant de remplir une ou plusieurs fonctions spécifiques avec des contraintes plus ou moins sévères telles que la consommation, la taille, les performances. . . A partir de ça on définit Linux embarqué comme étant une adaptation d'un noyau Linux dans un système embarqué, avec une diminution des fonctionnalités du noyau comme des espaces mémoire faibles ainsi que des services limités.

4. Distribution :

Une distribution Linux est un ensemble cohérent de logiciels rassemblant un système d'exploitation composé d'un noyau Linux et d'applications. Il existe plusieurs centaines de systèmes GNU+Linux, chacun ayant ses particularités, à usage automatisant et facilitant l'installation des composants nécessaires pour par exemple réaliser : pare-feu, routeur. . . Les distributions généralistes les plus populaires sont Debian, Gentoo, Mageia, Fedora, Slackware, OpenSUSE, ou encore Ubuntu permettant des usages variés. Certaines distributions sont orientées exclusivement vers l'entreprise avec un contrat de support annuel par exemple Red Hat Enterprise Linux, Ubuntu Long Term Support et SUSE Linux Enterprise.

5. Terminal ou client léger :

Un terminal intelligent ou client léger est un ordinateur qui fonctionne à partir de ressources stockées sur un serveur central au lieu d'un disque dur localisé. Les clients légers fonctionnent en se connectant à distance à un environnement informatique basé sur un serveur dans lequel la plupart des applications, des données sensibles et de la mémoire sont stockées.

Un client léger peut être utilisé de trois manières : services partagés, virtualisation de bureau ou basé sur un navigateur.

- (a) Avec les services de terminaux partagés, tous les utilisateurs des stations client léger partagent un système d'exploitation et des applications basés sur un serveur. Les utilisateurs d'un client léger à services partagés sont limités à des tâches simples sur leur machine, telles que la création de dossiers, ainsi que l'exécution d'applications.
- (b) La virtualisation des postes de travail, ou traitement de l'interface utilisateur, signifie que chaque poste de travail réside dans une machine virtuelle, qui est partitionnée des autres machines virtuelles sur le serveur. Le système d'exploitation et les applications ne sont pas des ressources partagées, mais ils vivent toujours physiquement sur un serveur distant. Ces ressources virtualisées sont accessibles à partir de tout appareil capable de se connecter au serveur.
- (c) Une approche basée sur un navigateur pour utiliser des clients légers signifie qu'un appareil ordinaire connecté à Internet exécute ses fonctions d'application dans un navigateur Web plutôt que sur un serveur distant. Le traitement des données est effectué sur la machine client léger, mais les logiciels et les données sont récupérés sur le réseau.

On peut dire que notre cas d'étude est une hybridation entre les "services de terminaux partagés" c-à-d que tous les utilisateurs des stations client légers partagent un système d'exploitation et des applications basées sur un serveur, et "l'approche basée sur un navigateur" qui consiste à effectuer le traitement des données sur la machine client léger, mais les logiciels et les données sont récupérés sur le réseau.

1.2 Approches pour personnaliser Linux

Lors de la création d'une distribution Linux, deux méthodes majeures s'imposent : Linux from scratch et création à partir d'une distribution existantes.

- **Linux From Scratch** : dans le cas où on est censé créer la distribution à zéro (c-à-d avoir un contrôle total et descendre jusqu'au niveau technique), LFS est le choix évident car il représente une suite d'instructions qui permet de créer un système linux personnalisé avec des règles précises et un contrôle total sur les paquets installés. Cette méthode demande de l'expertise et coûte cher en temps car il faut savoir ce qu'on fait à chaque étape et avec chaque commande.
- **Distro Based Linux (Linux à partir d'une distribution)** : dans ce cas, la création démarre déjà à partir d'une base qui est une distribution existante (Debian par exemple). Cela veut dire que les modules et les différentes parties nécessaires au démarrage du système d'exploitation sont déjà configurés, et de ce fait on aura moins de maîtrise de la distribution par rapport à LFS. Il est à noter qu'il existe plusieurs outils faciles à utiliser qui fournissent une large palette de personnalisations de la distribution base en peu de temps.

1.3 Étude de cas Linux

1.3.1 Pourquoi Linux

Sur les serveurs comme sur les postes de travail, les avantages de Linux et des logiciels libres par rapport aux solutions propriétaires établies, notamment Microsoft Windows, sont significatifs.

Les configurations proposées sont robustes et fiables. On obtient un système réactif et rapide, qui permet d'être productif sans se soucier des virus ou autres plantages. Il n'y a pas de frais de licence à payer, et le système est peu gourmand en ressources matérielles.

Il est aussi interopérable avec Microsoft Windows et Mac OS X, mais à la différence de ces deux concurrents propriétaires, toute information privée reste inaccessible et protégée de tout genre d'espionnage. En plus de tout cela, Linux est aussi gratuit et open source.

1.3.2 Structure de Linux

Cette partie présente une étude de la structure de linux et les différentes parties qui le compose. Nous allons aussi faire une étude sur les systèmes de fichiers supportés par linux.

— **Composants de linux :**

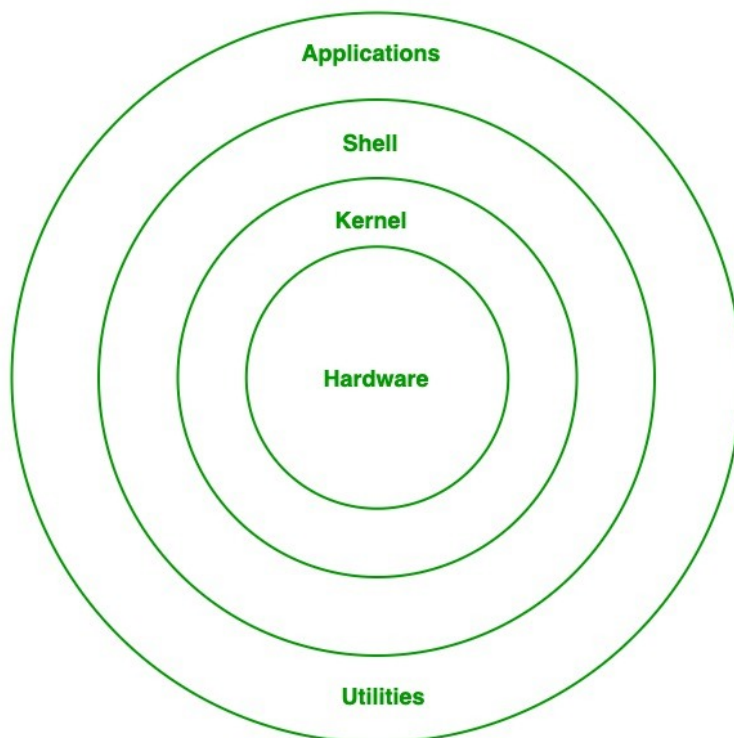


FIGURE 1.1 – Architecture de Linux

Linux, dans sa version basique, est composé principalement de trois composants :

1. **Kernel (noyau) :** Responsable de faire les importantes abstractions de systèmes d'exploitation comme la mémoire virtuelle et les processus.
2. **Les bibliothèques systèmes :** il s'agit d'un ensemble de fonctions qui assure l'interaction des applications avec le Kernel. Ces fonctions implémentent les fonctionnalités système qui ne nécessitent pas le privilège Kernel.

3. **Les utilitaires systèmes** : ce sont des programmes qui exécutent des tâches de gestion généralistes et spécialisées. Certaines de ces fonctions peuvent être appelées une fois pour initialiser et configurer certains aspects du système ; autres connus sous le nom de démons dans la terminologie UNIX pouvant s'exécuter en permanence et assurant des fonctions telles que les réponses à des connections, accepter des requêtes depuis les terminaux et mettre à jour les fichiers logs.

— **Le noyau linux approfondi :**

Le noyau linux forme le cœur du système d'exploitation Linux. Il fournit toutes les fonctionnalités nécessaires pour exécuter des processus, assure les droits d'accès et protège les ressources Hardware. Il implémente toutes les fonctionnalités pour se qualifier de OS, mais il manque certaines fonctionnalités qui sont assurées par les bibliothèques systèmes et les utilitaires.

1. **Architecture de Kernel :**

Le noyau linux a beaucoup évolué au fil des années, mais son architecture de base demeure plus ou moins inchangée. Il peut être décomposé selon les sous systèmes suivants :

- The Hardware abstraction layer : assure une virtualisation de matériel pour que les drivers soient portés dans n'importe quel matériel facilement.
- Memory manager : assure la gestion des ressources mémoire physique ainsi que la mémoire virtuelle.
- Scheduler : définit la politique d'ordonnancement des différentes tâches et processus.
- File system : permet de définir la façon de stocker les informations et de les organiser dans des fichiers
- IO subsystem : fournit une interface simplifiée et uniforme pour les périphériques.
- Networking subsystem : parmi les points forts de linux, il assure le support de différents protocoles réseaux.
- IPC (Inter-Process Communication) : gère les communications inter-processus.

2. **Modules Kernel** : Le Linux Kernel a la possibilité de charger des modules Kernel à la demande. Ces modules sont exécutés dans le mode kernel privilégié. Les modules kernels possèdent 3 composant :

- Module Management : permet aux modules de se charger en mémoire et communiquer avec le reste des modules.
- Le Driver Registration : permet d'annoncer à d'autres modules qu'un nouveau driver est disponible.
- A Conflict Resolution Mechanism : permet aux différents pilotes de périphériques de réserver des ressources et les protéger.

— **Les systèmes de fichiers :**

L'étude des systèmes de fichier est importante dans notre cas, le but derrière cela est de bien choisir un système qui permet d'optimiser notre distribution sur des machines bien spécifiques et selon des contraintes bien particulières.

Pour définir un système de fichiers ou système de gestion de fichiers (SGF), il s'agit de la façon dont les informations sont stockées et organisées dans des fichiers sur des supports de stockage. Parmi les systèmes existants nous pourrions citer :

1. **Ext** : signifie Extended File system, il est créé spécialement pour linux et il a 4 versions au moment actuel :
 - La première version n'est plus utilisée.
 - Ext2 est un système de fichier dit non journalisé donc non résistant aux d'arrêts imprévus mais cette caractéristique le rend utile pour les mémoires flash comme les usb.
 - Ext3 : c'est la version journalisé de Ext2 et il est compatible avec ce dernier, ce qui rend le passage de l'un à l'autre possible sans formatage.
 - Ext4 : une version plus sophistiquée qui réduit la fragmentation des fichiers. c'est la version la plus moderne de Ext et elle est aussi compatible avec les deux anciennes versions
2. **ReiserFS** : Contrairement à ext3, ReiserFS fut conçu comme un système de fichier journalisé. il est très efficace dans le cas de partitions de grandes tailles gérant un nombre de petits fichiers.
3. **BtrFS** : B-Tree File System, il est conçu pour être le successeur de ext. il offre une gestion de plusieurs disques et une compression transparente des données.
4. **JFS** : (Journaled File System) Il est caractérisé par une faible utilisation de processeur et de bonnes performances. Les partitions JFS peuvent être redimensionnées dynamiquement mais pas réduite. Il n'a pas été aussi testé sur des serveurs linux car il a été conçu pour le système d'exploitation AIX.
5. **XFS** : principalement développé pour l'OS SGI IRIX puis porté sous linux. Il est similaire à Ext4. Il a de bonnes performances sur les grands fichiers mais moins bons sur les petits fichiers.

Un système de fichier peut aussi désigner l'organisation des fichiers dans un système. Pour linux il s'agit d'une structure hiérarchique contenant des répertoires pouvant être omis ou non selon le cas d'utilisation de système : les répertoires tels que /home, /mnt, /opt et /root, peuvent être omis car ils ne sont pas indispensables pour le fonctionnement du système. Il est toujours possible de réduire le système de fichiers racine en supprimant /tmp et /var, mais ces omissions peuvent compromettre le fonctionnement de certains programmes. Ça sous-entend que les répertoires restants sont primordiaux pour une distribution (/etc,...).

1.3.3 Architectures Matérielle supportés par Linux

Linux fonctionne sur un nombre toujours croissant d'architectures de machines, qui ne sont pas toutes utilisées dans des configurations embarquées. Parmi ces architectures, nous en couvrirons 3 qui sont utilisées dans les systèmes Linux embarqués : ARM, Intel x86, MIPS.

- **ARM** : Advanced RISC Machine, est une famille de processeurs maintenus et promus par ARM Holdings Ltd. C'est une architecture très populaire dans plusieurs domaines d'application, et très fiable pour l'embarqué par rapport à son ensemble d'instructions qui est assez réduit. Présentement, Linux soutient environ 10 différentes ARMs, et 16 flatteries différentes. Selon la statistique de VDC (Virtual Device Context) en 2004, l'architecture ARM est la plus utilisée dans l'embarqué. On abordera cette partie plus en détail dans le prochain chapitre.
- **X86** : Intel est toujours le plus grand distributeur des processeurs de cette famille et quelques constructeurs aussi donnent leurs produits de famille x86. Elle est la plus populaire, linux supporte alors complètement l'architecture de x86. Dans le monde de Linux,

il y avait plein de distributions supportant x86. C'est un grand avantage pour la décision du choix d'une plateforme. Malgré le grand support, il représente une petite partie du marché de systèmes embarqués. Dans la plupart des cas, les concepteurs préfèrent les processeurs ARM, MIPS, et PowerPC à i386 en raison de la complexité et du coût.

- **MIPS** : Microprocessor without Interlocked Pipeline Stages, c'est une architecture de microprocesseur de RISC développée par MIPS Computer Systems. Le portage initial de Linux vers MIPS a été principalement effectué pour prendre en charge les stations de travail basées sur MIPS, bien que ceux-ci soient maintenant en grandes parties obsolètes (il n'y a plus de postes de travail MIPS en cours de fabrication). Il y a 2 types de MIPS. La puce à 32 bits, et une autre de 64 bits. Toutes les deux sont déjà portées en Linux Embarqué par Debian et RedHat.

1.4 Etude des processeurs ARM

1.4.1 La relation entre le système d'exploitation et l'architecture du processeur

Tout système d'exploitation dépend au moins d'un élément matériel : le processeur. Il existe différents processeurs, chacun fonctionnant différemment et ayant une "langue native" différente. Puisqu'un OS est "juste un programme" qui doit s'exécuter sur le CPU, il doit être écrit dans la langue native du CPU qui en dépend donc. On ne peut pas exécuter un Windows normal sur un processeur ARM ou PowerPC, par exemple. Il ne fonctionne que sur des processeurs compatibles Intel.

Les systèmes d'exploitation UNIX comme Linux, FreeBSD sont un bon exemple pour les systèmes d'exploitation qui peuvent être compilés pour différents processeurs et exécutés dessus.

1.4.2 Architecture de processeur ARM

L'architecture du processeur ARM est créée par Advanced RISC Machines, d'où le nom ARM, utilisé en particulier dans les appareils portables en raison de sa faible consommation d'énergie et raisonnable performance. Cela nécessite très peu de jeux d'instructions et de transistors. Il a une très petite taille. C'est la raison pour laquelle il convient parfaitement aux appareils de petite taille comme les smartphones, Raspberry Pi, les tablettes,...etc et il peut être appliqué à diverses conceptions telles que les périphériques 32 bits et les systèmes embarqués.

Les principales caractéristiques du processeur ARM sont :

- Systèmes multiprocesseurs
- Mémoire étroitement couplée : il a une faible latence
- Il possède une unité de gestion et de protection de mémoire
- Temps d'exécution d'un cycle par instruction
- Architecture en pipeline
- Grand nombre de registres

La principale différence entre l'architecture ARM et les autres architectures (CISC, pipeline, superscalaire...etc) tient dans les choix de conception de leurs jeux d'instruction : l'ARM est une

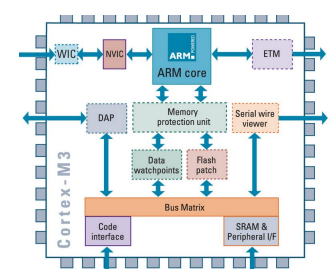


FIGURE 1.2 – Architecture du processeur ARM

architecture RISC (Reduced ISC) c-à-d ne supportent que des instructions simples et de taille fixe (4 octets pour le jeu d'instructions ARM standard), tandis que la plupart des architectures sont des architectures CISC (Complex Instruction Set Computer) c-à-d elles proposent des instructions nécessitant plus de cycles que d'autres, pour réaliser certaines tâches complexes. .

1.4.3 Système embarqué pour ARM

Un système embarqué est un système intelligent qui est conçu dans un but précis, contenant un petit ordinateur (processeur) qui pourrait être un processeur ARM. Ce dernier prend de plus en plus d'importance dans la conception de ce système car il nécessite moins de transistors, ce qui permet une taille de puce plus petite pour les circuits intégrés.

1.4.4 Raspberry Pi

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM, il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux, notamment Debian, et des logiciels compatibles.

Raspberry Pi dispose d'un processeur monocœur ARMv6 700 MHz, d'un GPU VideoCore IV et 512 Mo de RAM. Il utilise une carte SD pour son système d'exploitation et son stockage de données. Il prend en charge Raspbian, un système d'exploitation Linux léger basé sur Debian.

1.4.5 Contexte d'utilisation de l'architecture ARM

Pour rester compétitif, ARM propose une variété de styles ou de séries de cœurs de processeur. Certains sont commercialisés pour une variété de cas d'utilisation comme :

1. Un amplificateur audio numérique, un processeur vidéo numérique, un microcontrôleur pour un système d'extinction d'incendie ou un moniteur de fréquence cardiaque sophistiqué (Cortex-A).
2. Des modems 4G LTE et 5G, dans lesquels le temps est un facteur essentiel pour parvenir à la modulation (Cortex-R).
3. Des systèmes de contrôle et de freinage automobiles, les appareils photo numériques haute définition avec reconnaissance d'image, la synthèse sonore, la reconnaissance vocale et le radar (Cortex-M).
4. Des applications pouvant impliquer l'apprentissage automatique ou une autre forme de traitement de réseau neuronal. Elle comprend des routines utilisées pour la reconnaissance d'images et de formes, ainsi que dans l'apprentissage automatique (Ethos-N).
5. Un coprocesseur, en particulier en conjonction avec Cortex-A (Ethos-U). Des cœurs plus applicables dans les serveurs et les centres de données qui les hébergent (Neoverse).
6. Des cartes à puce, la certification USB et les applications de sécurité intégrées (Secure-Core).

ARM Architecture: For Diverse Embedded Processing Needs

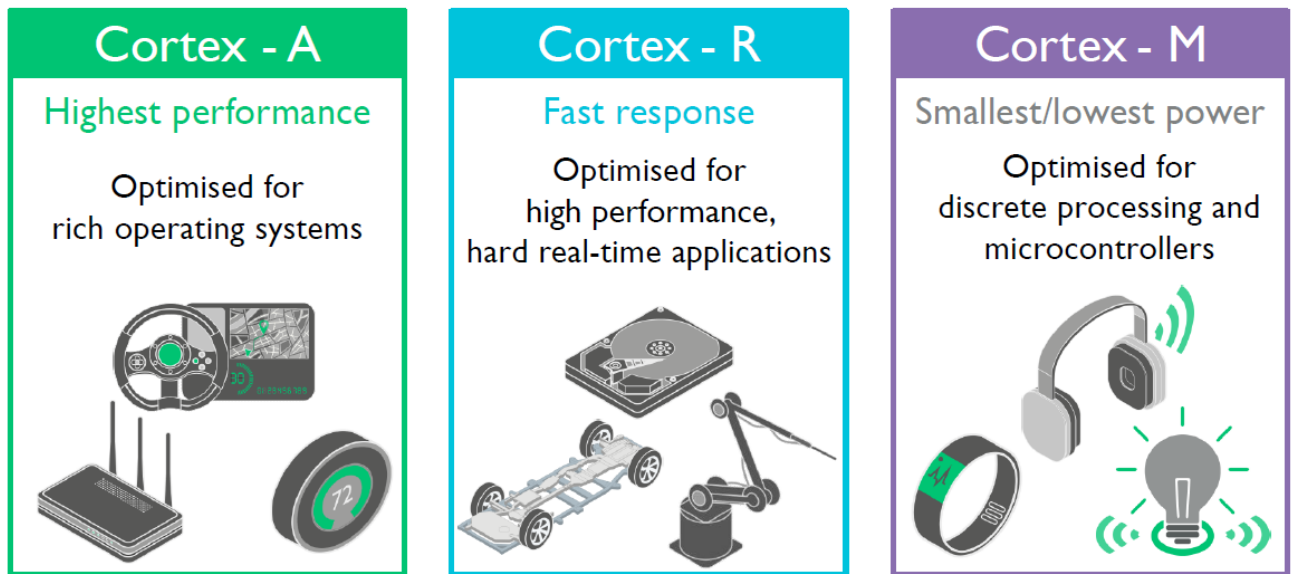


FIGURE 1.3 – Quelques contextes d'utilisation de l'ARM

1.5 Distributions Linux orientées enseignement

1. Edubuntu :

Edubuntu est un système d'exploitation basé sur la distribution GNU/Linux Ubuntu comprenant un ensemble de logiciels libres dédiés à l'éducation. Il contient des logiciels déjà existants dans Ubuntu comme : LibreOffice, Mozilla FireFox, une partie de la suite éducative KDE (Kdeedu) : Kalzium, Kanagram et des logiciels éducatifs installables par le moyen d'internet comme Celestia.

Spécifications :

- Applications utiles pour les classes dans les écoles primaires et secondaires voire des niveaux supérieurs
- Permet la réutilisation des vieux ordinateurs en clients légers
- Ne nécessite pas une connexion internet permanente.
- Parmi les applications qu'offre cette distribution : Apprentissage des langues, mathématiques, informatique, jeux éducatifs, utilisation du clavier, programmation, géographie, dessin, chimie, musique, physique...

2. Skolelinux :

Il s'agit d'un projet open source créé par des développeurs de la distribution Debian puis repris par une équipe norvégienne. Cette distribution peut être utilisée pour installer un serveur ou un poste de travail, et elle fournit des logiciels pour les écoles.

Spécifications :

- Distribution basée sur debian
- Trois modes d'installation : serveur, serveur d'application, station de travail

- L'avantage est que la communauté Debian est grande donc on peut utiliser tout les logiciel
- Permet aux étudiant d'accéder à leurs espace de travail en utilisant un user et password
- Offre des solutions de supervision et de contrôle parental
- Il permet de partager les ressources
- Solution complète qui assure la gestion complète de l'infrastructure (serveur, poste de travail, client léger, supervision,...)

3. The School Linux Project :

Le Linux Schools Project ou TLSP (anciennement Karoshi) est une distribution Linux basée sur Ubuntu conçue en 2000 dans le but de faciliter l'adoption de Linux pour les écoles du Royaume-Uni. Le projet gère deux distributions personnalisées :

- **Distribution du serveur** : comporte les fonctionnalités suivantes :
 - Contrôler le domaine principal : authentication, services de fichiers, services d'impression.
 - Assure une gestion des systèmes hétérogènes utilisant Windows
 - Package d'apprentissage en ligne : Moodle et plusieurs systèmes de gestion de contenu des sites Web, permettant des fonctionnalités complètes de calendrier et de messagerie.
- **Distribution du client** : La première version de Karoshi Client était basée sur PCLinuxOS. En juin 2012, le travail a commencé sur Karoshi Client version 2, qui aurait une interface plus proche de Gnome 2 que de Windows. Karoshi Client contient de nombreuses applications jugées nécessaires pour son fonctionnement, comme :
 - Un logiciel de production multimédia ainsi que des outils de programmation et des logiciels de visualisation.
 - Xfce est utilisé comme gestionnaire de bureau, avec un thème et une disposition de panneaux personnalisés.

4. Poseidon Linux :

Poseidon Linux est un logiciel open source destiné à la communauté scientifique internationale (étudiants, techniciens, professeurs, chercheurs). Ce système d'exploitation est basé sur Ubuntu et inspiré de Quantian Linux. Il propose plusieurs outils spécifiques dans les domaines de la bathymétrie, de la cartographie des fonds marins, du SIG et de la visualisation 3D. Les versions 32 bits et 64 bits ont été publiées. Cette version était basée sur le support à long terme (LTS) Ubuntu 10.04, car ses objectifs sont la stabilité, la convivialité et la prise en charge de l'ensemble du système lorsqu'il est utilisé professionnellement dans les universités, instituts, collèges et également à la maison. Le nom Poséidon ,qui veut dire le Dieu des mers dans la mythologie grecque, a été choisi car un grand nombre d'océanologues ont été impliqués dans le développement du système.

Spécifications :

- Il contient les langages de programmation Fortran, Kile et Lyx pour l'écriture scientifique, la modélisation numérique, la visualisation 2D / 3D / 4D, les statistiques, la génétique, la bio -informatique ...etc.
- Des programmes quotidiens courants sont également inclus, tels que LibreOffice, des navigateurs Web, des packages multimédias et certains jeux.

- Le cœur de Poséidon est toujours la famille Debian en raison de la plus grande stabilité et de la plus grande quantité de logiciels disponibles dans les sites de dépôt.

5. **UberStudent :**

UberStudent, appelé aussi “Linux for learners” est un système d’exploitation gratuit créé en 2010 qui vise comme utilisateurs principaux : les étudiants, les enseignants et les établissements scolaires et il est aussi facile à utiliser pour les débutants en Linux que les utilisateurs lambda. Il utilise aussi le fameux bureau Xfce.

Spécifications :

- UberStudent est livré avec la plupart des logiciels habituels tels que LibreOffice, Chromium et Firefox pour les tâches quotidiennes.
- UberStudent regroupe des applications spécialement sélectionnées telles que Freeplane qui permet un brainstorming facile d’idées et de pensées. Il existe le très populaire gestionnaire de références Zotero ainsi que d’autres logiciels tels que GIMP pour la création et l’édition de photos et d’images.
- Il contient également des sous-menus tels que Livres, Sujets, Données, Recherche et écriture ,et Gestion du temps.
- On peut également ajouter facilement des logiciels éducatifs supplémentaires en accédant aux biens nommées «Applications pédagogiques supplémentaires».
- UberStudent contient également des jeux soigneusement sélectionnés. Ces jeux aideront à libérer le stress des études et aideront également à entraîner le cerveau dans le processus. Des jeux basés sur la logique tels que Cut the Rope, Puzzle Collection, Sudoku, Wonderputt.

Chapitre 2

Analyse des besoins

Comme le but de ce projet est de créer une distribution Linux pour un niveau scolaire spécifique qui est le Lycée, une étude des besoins s'avère nécessaire avant d'attaquer la réalisation. A travers les exigences techniques spécifiés dans le cahier des charges et les besoins métiers exigés par l'environnement pour lequel cette distribution est destiné, nous avons pensé à organiser ces besoins de la façon suivante :

2.1 Besoins techniques :

Afin d'assurer la conformité de la distribution ainsi que son bon fonctionnement sur les terminaux (qui s'avèrent être des machines très faibles en termes de puissance) cette dernière doit être fonctionnelle sur tout type de machine quel que soit son processeur et ses performances. Ceci dit, elle doit être optimisée sur tous les niveaux : niveau noyau, niveau utilitaires système et niveau interface graphique.

2.2 Besoins fonctionnels :

- Assurer l'accès à distance au serveur, à travers les protocoles RDP et SSH.
- Assurer la séparation des espaces utilisateurs
- Assurer l'accès aux différentes ressources sur internet ou en local.
- Assurer des applications de bases pour lectures vidéos et traitement de texte.
- Assurer l'ergonomie, la sécurité, le confort et la simplicité de l'utilisation de la distribution.

2.3 Besoins utilisateurs :

- Mettre à la disposition des utilisateurs (lycéens) les différents supports de cours, les travaux dirigés et pratiques ainsi que les prochains devoirs à remettre.
- Mettre en place des logiciels d'apprentissage pour les différentes spécialités et matières (mathématiques, physique, sciences de la vie, langues...).
- Mettre en place des simulateurs et des applications de test de connaissances afin de permettre un suivi de l'élève.

Chapitre 3

Réponses aux besoins

Afin de répondre aux besoins cités dans le paragraphe précédent et trouver un compromis entre toutes les exigences, les contraintes et éventuellement les alternatives offertes par des solutions existantes, nous avons fait les études comparatives suivantes :

3.1 Étude comparative des outils existants

Les outils permettant de personnaliser une distribution Linux sont nombreux. Chaque outil offre des possibilités meilleures qu'un autre cela nous a mené à faire une étude comparative de ces derniers afin de choisir le mieux adapté à nos besoins :

3.1.1 Live Build :

Live Build est un ensemble de scripts pour créer des images systèmes live (Debian / Ubuntu / Linux Mint ... netboot, des images iso, des images usb et quelques autres). L'idée derrière Live Build est une suite d'outils qui utilise un répertoire de configuration pour automatiser et personnaliser complètement tous les aspects de la création d'une image live.

Il est très flexible car il permet de décider, par exemple, du type de noyau qu'on souhaite intégrer à notre système et à quoi devrait ressembler notre bureau par défaut.

TABLE 3.1 – Avantages et inconvénients de Live Build

Avantages	Inconvénients
<ul style="list-style-type: none"> • Dans l’environnement stable Debian, live build a toujours été solide et performant. • Facile à utiliser (via des commandes) • Permet de créer une distribution personnalisée • Peut réduire la taille de l’image • Permet de créer un LiveCD qui permet de tester ou de faire tourner une distribution GNU/Linux sur un ordinateur sans altérer le disque dur, puisque le système n’utilise que la RAM 	<ul style="list-style-type: none"> • Ne tolère pas les erreurs au cours de la construction • nécessite de rebuild le tout à chaque fois qu’on change quelques choses dans la configuration ce qui prend beaucoup de temps

3.1.2 Linux Live Kit :

Linux Live Kit est un ensemble de scripts qui permettent de créer une version linux Live depuis une distribution déjà installée et cette distribution sera bootable depuis un CD ou un usb. Il est connu pour sa simplicité d’utilisation. En effet,

TABLE 3.2 – Avantages et inconvénients de Linux Live Kit

Avantages	Inconvénient
<ul style="list-style-type: none"> • Simple à utiliser, une seule instruction pour construire l’iso • Installation des modules et personnalisation via apt-get • Possibilité de supprimer des modules 	<ul style="list-style-type: none"> • Il génère seulement un live CD ou USB bootable • Il n’offre pas de moyens pour l’adapter sur d’autres architectures

3.1.3 Pinguy builder :

Pinguy Builder est un fork de Remastersys créé par le développeur de la distribution Pinguy OS et est devenu un simple et facile à utiliser utilitaire graphique.

Pinguy Builder permet de créer une sauvegarde complète du système (y compris les données de l’utilisateur) ou une ISO distribuable (sans les données personnelles) de ses fichiers systèmes et de son interface utilisateur.



FIGURE 3.1 – Logo de Pinguy Builder

TABLE 3.3 – Avantages et inconvénients de Pinguy builder

Avantages	Inconvénients
<ul style="list-style-type: none"> • Permet plusieurs configurations et personnalisations • Permet de publier l'image de marque en utilisant notre propre logo, theme, menu de boot,...etc • Simple et facile à utiliser avec une interface graphique 	<ul style="list-style-type: none"> • L'image ISO peut dépasser 4Go ce qui est contre la norme ISO 9660 qui ne permet pas des fichiers plus volumineux • Une fois la compilation commence on ne peut pas l'arrêter ou faire une pause, il faut tout refaire du début • Nécessite des paquets et des dépendances pour obtenir une image utilisable (par ex.pour compresser,...)

3.1.4 Slax :

Slax est un système d'exploitation Live basé sur Linux. L'ensemble du système d'exploitation Slax réside dans un seul répertoire /slax/ sur votre appareil, ce qui facilite l'organisation avec vos autres données.

Slax fournit un gestionnaire de fenêtres FluxBox déjà pré configuré pour les tâches les plus courantes. Il inclut un navigateur Web chrome, un émulateur de terminal xterm, un simple éditeur de texte et un calculateur de feuille de calcul.



FIGURE 3.2 – Logo de Slax

TABLE 3.4 – Avantages et inconvénient de Slax

Avantages	Inconvénients
<ul style="list-style-type: none"> • Logiciel gratuit. • Compatible avec tous les processeurs Intel et architecture AMD (32 et 64bits). • 512MB de mémoire RAM minimale. • Facile à personnaliser avec beaucoup de librairies déjà existantes. 	<ul style="list-style-type: none"> • Ne peut plus personnaliser après avoir généré l'image. • L'image sera générée sur CD/DVD. • Ne génère pas un ISO installable mais juste un CD Live.

3.1.5 Ubuntu Imager :

Ubuntu imager est un projet de Distroshare.com, qui est un site web qui vise à partager des distributions personnalisées open source de systèmes d'exploitation.

Distroshare ubuntu imager permet de créer un installable Live ISO à partir d'un système Ubuntu installé.



FIGURE 3.3 – Logo de Ubuntu Imager

TABLE 3.5 – Avantages et inconvénients de Ubuntu

Avantages	Inconvénients
<ul style="list-style-type: none"> • Simple et facile à utiliser, il suffit de lancer un script shell. • La création de l'image ISO prend peu de temps par rapport à d'autres outils. • Il permet d'apporter beaucoup de modifications et personnalisations à la distribution. 	<ul style="list-style-type: none"> • Petite communauté. • Absence de documentation détaillée. • La taille de l'image ISO peut devenir très volumineuse puisqu'elle dépend du système ubuntu qu'on cherche à copier

3.1.6 Récapitulatif

Criteres	Live Build	Linux Live Kit	Pinguy Builder	Slax	Ubuntu Imager
Documentation existante	++	++	+	++	-
Facile à utiliser	++	+++	++	++	++
Possibilité de faire plusieurs personnalisations	+++	+	++	++	++
Rapide en termes d'exécution et de génération des images	+	+	-	+	+
Génère des images de tailles acceptables	++	+	-	-	-
Possibilité de générer différentes architectures	+++	-	-	+	-

D'après ce tableau comparatif, on déduit que Live Build est meilleur que les autres outils. En effet :

- Il est largement documenté et il existe plusieurs tutoriels pour son apprentissage.
- Il est facile à utiliser du moment qu'on utilisera seulement 2 commandes pour la création de la distribution et la spécification de toutes les options (architecture, ...).

- Il est possible de personnaliser la distribution en installant les packages voulus avec la commande apt.
- Il génère des images de tailles minimales.
- Il permet de spécifier plusieurs types d'architecture (ARM, 32bits, 64bits...etc).

C'est pour cela qu'on a opté pour **Live Build**.

3.2 Étude comparative des applications :

Une grande variante d'applications qui répondent aux besoins fonctionnels et aux besoins utilisateurs spécifiés dans le paragraphe précédent existe. Mais les exigences et les contraintes nous mène à choisir minutieusement ces applications. Les paragraphes suivants représentent des comparaisons des outils, dans chaque catégorie, les plus connus et proches en termes de réponses aux exigences, afin de pouvoir choisir ceux les plus convenables.

3.2.1 Environnement de bureaux :

TABLE 3.6 – Avantages et inconvénients des différents environnement de bureaux

	LXDE	XFCE	GNOME
Avantages	<ul style="list-style-type: none"> — L'un des environnements de bureau les plus légers du marché. — Simple et minimal. 	<ul style="list-style-type: none"> — Complet en termes de fonctionnalités. 	<ul style="list-style-type: none"> — Le plus connu et le plus utilisé. — Permet des fonctionnalités avancées.
Inconvénients	<ul style="list-style-type: none"> — S'appuie sur de nombreuses applications d'autres environnements de bureaux (Desktop Environment). 	<ul style="list-style-type: none"> — Exige une machine plus puissante pour pouvoir fonctionner confortablement (jusqu'à 450 Mo de RAM). 	<ul style="list-style-type: none"> — Nécessite beaucoup de RAM. — Pas possible de changer l'économiseur d'écran, le thème système ou le pack d'icônes.

Afin de répondre aux exigences techniques et spécifications du Hardware, LXDE est petit en taille et permet les fonctionnalités nécessaires donc mieux adapté.

Une autre possibilité d'optimisation côté système, pour gagner un peu d'espace et de performances, les drivers concernant les appareils suivants peuvent être enlevés :

- Imprimante

- Scanner
- Modem, fax

Comme les lycéens n’auront pas besoin de les utiliser, ils vont résider dans le système sans utilité.

Il existe aussi des répertoires contenant l’éditeur de texte de la ligne de commande “nano” et les manpages (documentation des commandes) et pouvant être enlevés vu qu’ils ne seront pas utilisés.

3.2.2 Accès à distance :

TABLE 3.7 – Avantages et inconvénients des différents outils d’accès à distance

	Remmina	OpenSSH	RealVNC
Avantages	<ul style="list-style-type: none"> — Bonne performance. — Flexibilité de connexion. — Plusieurs protocoles : RDP, VNC, SSH, EXEC. 	<ul style="list-style-type: none"> — Sécurité. 	<ul style="list-style-type: none"> — Multi-plateforme. — Facile à utiliser.
Inconvénients	<ul style="list-style-type: none"> — Manque de fonctionnalités supplémentaires. 	<ul style="list-style-type: none"> — Nécessite des paquets et des dépendances supplémentaires ce qui augmente la taille. 	<ul style="list-style-type: none"> — Ne permet pas des sessions multiples. — Ne permet pas le transfert de fichiers.

Après avoir fait la comparaison la dessus, **Remmina** est le plus approprié à cette distribution à cause de sa simplicité, son support pour plusieurs protocoles et ses bonnes performances.

3.2.3 Serveur vidéo et client :

TABLE 3.8 – Avantages et inconvénients des outils vidéos

	VLC Media Player	Kodi
Avantages	<ul style="list-style-type: none">— Robuste.— Contient les fonctionnalités nécessaires.— Serveur et client.	<ul style="list-style-type: none">— Facilité et simplicité d'utilisation.— Permet la gestion et organisation des médias.
Inconvénients	<ul style="list-style-type: none">— Manque de bibliothèques et de fonctionnalités avancées de lecteur de musique.	<ul style="list-style-type: none">— Problème avec les “addon” (modules complémentaires).

Comme il offre les fonctionnalités basiques nécessaires (taille réduite) et travaille côté serveur et client, **VLC Media Player** est plus approprié à cette distribution.

3.2.4 Logiciels bureautiques :

TABLE 3.9 – Avantages et inconvénients des logiciels bureautiques

	LibreOffice	WPS Office	Apache Open Office
Avantages	<ul style="list-style-type: none">— Fonctionnalité complète.— Compatibilité avec les formats de fichiers MS Office.	<ul style="list-style-type: none">— Outils d'édition de PDF.— Belle interface utilisateur.	<ul style="list-style-type: none">— Utilise la licence Apache.— Liste complète des langues prises en charge.
Inconvénients	<ul style="list-style-type: none">— Design obsolète.	<ul style="list-style-type: none">— La version gratuite contient des publicités.	<ul style="list-style-type: none">— Prise en charge limitée des formats de fichiers MS Office.

Le choix de **LibreOffice** est un bon choix pour la distribution, car malgré son design obsolète (peu important en comparant avec les performances et l'espace), il offre de grandes fonctionnalités et une meilleure compatibilité.

3.2.5 Navigateur Web :

TABLE 3.10 – Avantages et inconvénients des Navigateurs Web

	Netsurf	Midori	Mozilla
Avantages	<ul style="list-style-type: none">— Multi-plateforme écrit en C.— Dispose de son propre moteur de mise en page.— Prend en charge HTTPS.	<ul style="list-style-type: none">— Interface sans encombrement et facile à utiliser.— Prend en charge HTML5, la gestion des onglets et des fenêtres.— Possède un correcteur orthographique et supporte les extensions.	<ul style="list-style-type: none">— Parmi les navigateurs les plus rapides.— Contient des options avancées.
Inconvénients	<ul style="list-style-type: none">— Pas d'inconvénients jusqu'à présent.	<ul style="list-style-type: none">— Faible nombre d'extensions disponibles.— Pas d'avertissement ou de chiffrement de certificat TLS.	<ul style="list-style-type: none">— Consomme beaucoup de mémoire.

Netsurf Netsurf est le mieux approprié à cause de son minimalisme, sa rapidité et sa consommation réduite de ressources

3.3 Applications utilisateur

Pour les applications concernant les spécialités spécifiques au lycée, on trouve différentes matières :

- Chimie :
gElemental : un visionneur de tableaux périodiques qui fournit des informations détaillées sur les éléments chimiques. Il offre une interface utilisateur améliorée, un ensemble de données mis à jour, un meilleur format de données, une vue de liste et une coloration plus facile.

Il se différencie de ses concurrents (comme Kalzium) en étant plus basique et moins sophistiqué donc convenable aux lycéens.

- **Math :**

GeoGebra : application destinée à l'apprentissage et à l'enseignement des mathématiques et des sciences, Il permet de :

- Résoudre les fonctions mathématiques
- Résoudre des calculs, des problèmes numériques et statistiques
- Tracer graphe à partir d'une fonction
- Résoudre un problème géométrique.

- **Géographie :**

Marble : est une visionneuse de carte open source, un explorateur d'atlas et une visionneuse de globe. Développé par l'équipe KDE. Marble permet de planifier des itinéraires et peut rechercher des cartes à partir de plusieurs sources. Il prend en charge les cartes en ligne et hors ligne.

- **Littérature et langues :**

Gnome-dictionary : un client DICT (Dictionary Network Protocol) écrit en C, parmi ses spécifications :

- Prend en charge plusieurs langues.
- Possibilité de partage.
- Interface simple et facile à utiliser.

- **Jeux :**

- **Sudoku :** le fameux casse-tête combinatoire de placement de nombres basé sur la logique.
- **Jeu de cartes :** Solitaire, le plus connu, le plus simple et le plus facile à utiliser.

Chapitre 4

Gestion Projet

L'équipe de travail “**DistroBuilders**” est une équipe spécialisée dans la création des distributions Linux orientées vers un domaine spécifique selon les besoins de ses clients. Parmi leurs projets les plus importants on trouve **Lykeio** (Lycée en Grec), une distribution orientée enseignement, pour les lycéens, et basée sur Debian.

Pour réaliser ce projet, l'équipe a été constituée (distribuée) comme suit :

- **Narimane** : Chef de projet, feedback le travail, coordonne entre les membres de l'équipe et prend en charge les réunions avec le client.
- **Nour** : Responsable de contrôle de qualité, assure la bonne qualité des livrables et des différents travaux en effectuant un workflow de vérification et de validation.
- **Nassim** : Consultant technique, fournit un soutien et des conseils techniques pour fonctionner plus efficacement et prendre les bonnes décisions.
- **Rezak** : Responsable Recherche et Développement, guide l'équipe de chercheurs dans leurs recherches et transmet les informations importantes concernant le produit final ainsi que les éventuelles contraintes et risques.
- **Amine** : Support technique, effectue des analyses sur les capacités matérielles, logicielles et réseau, améliore l'efficacité du système en fournissant des solutions innovantes.

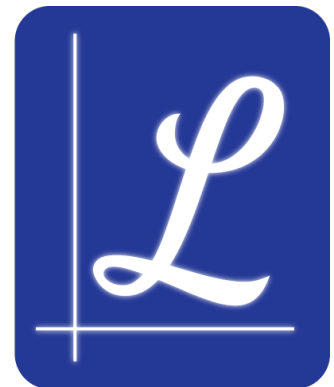


FIGURE 4.1 – Logo Lykeio

Certes le travail d'équipe donne de meilleurs résultats et une meilleure efficacité mais ceci nécessite une bonne organisation et une bonne communication, et une approche organisationnelle doit être suivie afin de s'assurer d'avoir le maximum revenu de nos efforts.

La démarche générale de gestion de notre projet se résume dans les points suivants :

1. Définition des objectifs du projet : comme mentionné dans la description du projet et à travers les réunions avec le client, nous avons pu faire sortir les objectifs principaux de ce projet qui sont documentés dans ce rapport.

2. Découpage du projet et définition de la liste des tâches : après avoir fait sortir les objectifs et afin de commencer le travail, nous avons fait une liste des tâches (initiale puis modifiée), en utilisant Google Sheets, pour répartir le travail et être le plus efficaces possible.
3. Ajout des durées et des contraintes sur les tâches : certaines tâches sont plus importantes ou urgentes que d'autres, pour cela nous avons trié les tâches selon leur ordre d'importance et selon des dates limites afin de pouvoir planifier des entrevues avec le client, discuter le travail fait et finir dans les délais prévus.
4. Planification et suivi du projet : Pour cela, nous avons réalisé un planning initial en utilisant l'outil InstaGantt, ce planning a subi plusieurs modifications à cause des contraintes rencontrées lors du travail :
 - La mauvaise compréhension du problème.
 - Un changement dans la priorité ou le niveau d'importance d'une certaine tâche (par exemple la tâche d'implémentation a été prévue pour la 6ème semaine mais au final il fallait prendre plus de temps pour prendre les bonnes décisions et les meilleurs choix donc ça a été décalée un peu.)
 - Les différents feedbacks donnés lors des entrevues.
 - Contrainte de temps (travaux en parallèle, CI ...etc).

Ces mises à jour se résument en :

- Création de nouvelles tâches.
- Extension des délais.
- Ré-ordonnancement des tâches.
- Réaffectation de certaines tâches.

Notre équipe tenait des réunions hebdomadaires (présentielles ou en ligne), et parfois plus qu'une réunion par semaine (selon la nécessité), pour discuter l'état d'avancement, planifier la semaine à venir, distribuer les tâches et discuter les problèmes rencontrés et les éventuelles solutions.

les PVs des réunions se trouvent en Annexe.

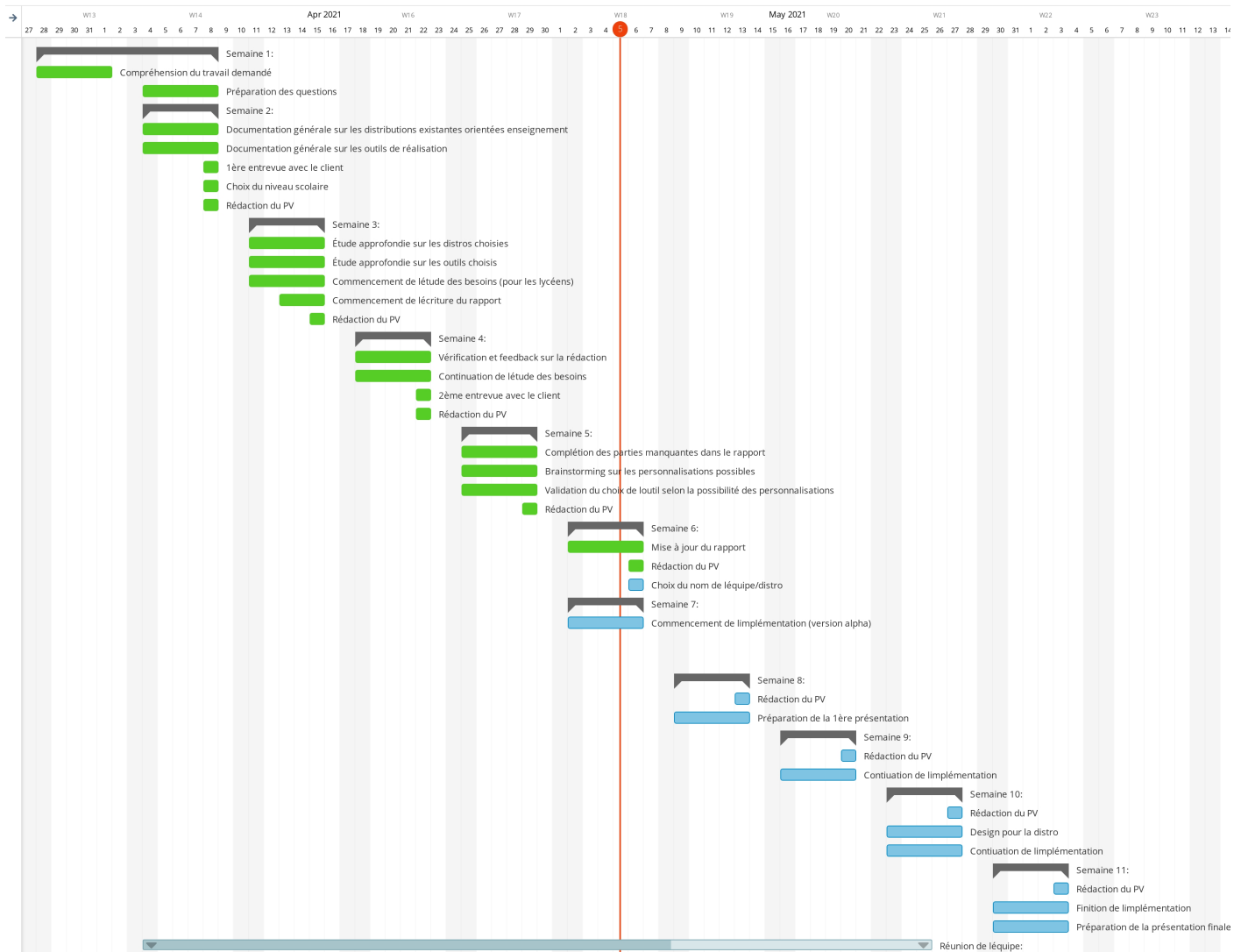


FIGURE 4.2 – Planning du travail

Note : Il existe des tâches (en bleu) non encore accomplies et non affectées.

Conclusion

A l'issue de la première partie, nous avons vu que l'utilisation d'une architecture basée sur des terminaux est une solution moins coûteuse que le renouvellement du parc informatique d'un établissement scolaire (lycée pour notre cas), ceci dit l'utilisation d'une distribution minimale orientée enseignement qui va aider les élèves à mieux s'organiser et apprendre rapidement et facilement, et aider l'administration aussi à optimiser les ressources disponibles.

Donc, nous sommes parvenus à faire une étude bibliographique sur les concepts de base du domaine d'étude de ce projet et sur les différentes distributions existantes orientées enseignement afin de comprendre leurs architectures ainsi que leurs différentes applications fournies, et c'est ce qui nous a permis de s'inspirer et de faire une première conception de notre distribution, tout en répondant aux exigences énoncées dans le cahier de charge.

Afin de réaliser notre distribution, nous avons fait une étude comparative des différents outils de réalisation. En se basant sur certains critères, nous avons pu choisir un outil qui permet de répondre, au maximum, aux exigences du CDC et d'effectuer des personnalisations. Créer une distribution à partir d'une distribution existante s'avère être une solution très efficace si le temps et la simplicité sont des critères de choix. De plus, ça nous permet de faire plusieurs personnalisations qui répondent aux besoins fonctionnels et non-fonctionnels déjà identifiés lors de l'étude des besoins effectuée.

Enfin, tout cela tourne autour d'un travail d'équipe qui doit être géré et bien organisé en appliquant les méthodes de gestion de projet.

Bibliographie

- [1] <https://www.techopedia.com/definition/28254/processor>.
- [2] <https://www.geeksforgeeks.org/arm-processor-and-its-features/>.
- [3] <https://www.infobidouille.com/pcworld/>.
- [4] <https://www.zdnet.com/article/arm-processors-everything-you-need-to-know-now/>.
- [5] <https://stackoverflow.com/questions/3790419/why-an-operating-system-is-called-as-3790514>.
- [6] <https://www.techopedia.com/definition/28254/processor>.
- [7] https://kadionik.vvv.enseirb-matmeca.fr/embedded/tipe-pham_viet_hung.pdf.
- [8] <http://meseec.ce.rit.edu/551-projects/spring2017/2-3.pdf>.
- [9] <https://edu.kde.org/>.
- [10] https://linuxhint.com/best_ms_office_alternatives_linux/.
- [11] <https://www.online-sciences.com/computer/midori-web-browser-advantages-and-disadvan>.
- [12] <https://www.maketecheasier.com/linux-education-apps-for-students/>.
- [13] <https://www.howtogeek.com/howto/33552/htg-explains-which-linux-file-system-should-y>.
- [14] <https://www.maketecheasier.com/lxde-vs-xfce/>.
- [15] <https://www.tomshardware.com/reviews/fedora-16-gnome-3-review,3155-11.html>.
- [16] Maurice J. Bach. *The design of the UNIX operationg system*. Pearson Education, 1986.
- [17] Karim Yaghmour Jon Masters Gilad Ben-Yossef and Philippe Gerum. *Building embedded Linux systems*. O'reilly, 2008.
- [18] Patrick Cegielski. *Conception de systeme d'exploitation, le cas Linux*. Eyrolles, 2004.
- [19] Pierre Ficheux. *Linux Embarqué 2eme édition*. Eyrolles, 2002.
- [20] P. Raghavan Amol Lad Sriram Neelakandan. *Embedded Linux system design and development*. Auerbach Publications, 2006.