الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد– تلمسان

Université Aboubakr Belkaïd-Tlemcen

كلية التكنولوجيا

Faculté de Technologie

Département de Génie Electrique et Electronique (GEE)

Filière : Electronique

MASTER INSTRUMENTATION

PROJET DE FIN D'ETUDES

Présenté par : FOUDI MOHAMED AMINE **et** ZEGGAI ASSIM

Intitulé du Sujet

Étude et réalisation pratique d'une gestion D'une salle d'attente

Soutenu en juillet 2021, devant le jury composé de :

Mr MASSOUM NORDINE	**MCB**	**Univ. Tlemcen**	**Président**
Mr NEMMICHE AHMED	**MCB**	**Univ. Tlemcen**	**Encadreur**
Mr MOULAY KHATIR NASSIM	**MCB**	**Univ. Tlemcen**	**Examinateur**

Année Universitaire 2020-2021

Remerciement

Louange à dieu le tout puissant, clément et miséricordieux de m'avoir permis de réaliser ce projet.

Nous remercions vivement :

Monsieur ***NEMMICHE AHMED*** *enseignant au département d'électronique, pour l'honneur qu'il nous a fait en nous encadrant, et son entière disponibilité durant toutes les étapes de notre projet.*

Nos remerciements vont aussi aux membres du jury, Présidé par Monsieur ***MASSOUM NORDINE*** *et examiné par Monsieur* ***MOULAY KHATIR NASSIM****.*

Nous tenons à leurs exprimer notre profonde reconnaissance pour l'honneur qu'ils nous ont fait en acceptant d'examiner notre travail.

Nous profitons de cette occasion pour témoigner notre grande reconnaissance au responsable du Master et professeur :

Monsieur BENAHMED NASRDINNE *pour son suivi, son enseignement et son aide pendant toute la formation de notre Master sans oublier sa contribué à la réalisation de ce projet.*

Nos vifs remerciements s'adressent également à tous les membres de nos familles qui nous ont aidés par leur soutien précieux.

Pour terminer, nous tenons à remercier tous nos collègues qui nous ont aidés et qui nous ont apporté tout le soutien moral nécessaire pour mettre à bien notre projet.

Liste des abréviations

RPi : Raspberry pi

WIFI : Wireless Fidelity

USB : Universel Serial Bus

RAM: Mémoire vive (Random Access Memory) TE: Terminal Equipment

Liste des figures

Chapitre I :

Chapitre II :

Chapitre III :

الملخص

يتحدث مشروعنا عن المشكلات التي واجهها الأشخاص قبل وجود نظام غرفة الانتظار ويقدم أيضًا طريقة لفهم كيفية عمل نظام غرفة الانتظار.

بعد ذلك، تعرفنا على ميزات Raspberry Pi وأنواعها والعديد من الاستخدامات ونظام التشغيل Raspbian.

بعد ذلك، استخدمنا بعض أوامر Linux الأساسية.

في الختام، تمكنا من برمجة مشروعنا من خلال Python باستخدام مكتبات TKinter و Socket وبعض الأساسيات

Abstract

Our project talks about the problems that people faced before the waiting room system existed, and also presents a way to understand the working of the waiting room system.

Next, we learned about the features of Raspberry Pi, its types, its many uses, and its Raspbian operating system.

Then we used some basic Linux commands.

In conclusion, we were able to program our project through Python using the TKinter and Socket libraries and some of the basics

Résumé

Notre projet décrit les problèmes auxquels les gens étaient confrontés avant l'existence des systèmes de gestion de la salle d'attente et présente également un moyen de comprendre le fonctionnement du système en question.

Par suite, nous avons découvert les fonctionnalités de Raspberry Pi, ses types, ses nombreuses utilisations et son système d'exploitation Raspbian.

Ensuite, nous avons utilisé quelques commandes Linux de base.

En conclusion, nous avons pu programmer notre carte via Python en utilisant les bibliothèques TKinter et Socket et autres librairies.

Introduction Générale

Dans notre vie de tous les jours, nous passons une grande partie de notre temps dans des files d'attente des bureaux de poste, des magasins, des hôpitaux et de nombreux autres endroits qui nous fournissent une variété de services. Trop de temps est perdu dans ce processus d'attente, un temps que nous pouvons passer à faire beaucoup d'autres choses utiles, mais ce temps est consacré à un processus où nous sommes incapables de faire quelque chose de plus bénéfique. Les clients et nous, en tant que société qui valorise le temps, considérons généralement le processus d'attente comme un travail sans valeur et si le processus d'attente dure trop longtemps, nous associons cette perte de temps à un service médiocre et à une mauvaise qualité de l'entreprise ou du service auquel nous sommes confrontés et les employés inoccupés ou les équipements inutilisés représentent des activités sans valeur ajoutée. Pour éviter ces situations ou les réduire à un temps d'attente minimal, la majorité des entreprises ont mis en place des processus d'amélioration continue dont le but ultime est d'éliminer toute forme de gaspillage, y compris l'attente. Tous ces exemples démontrent l'importance de l'analyse des files d'attente. S'ils constituent une nuisance incontournable, ils respectent néanmoins une certaine équité pour plus d'efficacité : premier arrivé, premier servi

Notre étude de projet propose une des nombreuses manières d'organiser ces services. Elle consiste à mettre en place un système de communication pour gérer le service dans une salle d'attente équipée d'un distributeur de tickets basé sur un microcontrôleur.

- ✓ **Le premier chapitre** décrit la salle d'attente, et formule le problème à résoudre à travers les étapes pratiques auxquelles nous participons dans notre recherche.
- ✓ **Le deuxième chapitre** se concentre sur une description détaillée de la carte Raspberry Pi et comment la programmer via Python sur linux.
- ✓ **Le troisième chapitre** est la mise en œuvre de la programmation, puis la réalisation pratique.

Chapitre I

Généralité

I.1 Introduction :

Depuis la plus haute antiquité les hommes ont cherché un moyen de favoriser la gestion des personnes en milieu professionnel tel que les hôpitaux, les aéroports, les postes et les différents systèmes de service. Pour cela une quantité importante de travaux a pris une nouvelle dimension dont le but est de réaliser des systèmes de gestion des personnes.

I.2 Historique :

Au fil du temps, l'homme cherche à se développer et à améliorer sa vie, et ceci dans le but de réduire les dépenses et de gagner du temps. On peut citer à titre d'exemple la salle d'attente (figureI.1). C'est pourquoi les scientifiques ont mis au point des théories pour éviter ce service médiocre, et éliminer toutes les formes de gaspillage, ou encore pour augmenter l'efficacité et d'améliorer le service : premier arrivé, premier servi.

Figure I.1 : salle d'attente d'un hôpital en 1950

I.3 Notions de base

3.1 Description d'une salle d'attente :

Une salle d'attente (figureI.2) est une salle à la disposition des utilisateurs dans un bâtiment pour leur permettre d'attendre le moment où ils seront reçus.

L'espace de cette salle peut contenir divers équipements (figure I.3) tels que des sièges et une table basse sur laquelle sont placés des magazines et des journaux, permettant aux utilisateurs

de se plonger dans la lecture pendant qu'ils attendent, tandis que des boîtes de jeux pour enfants sont également disponibles et de la musique de fond peut être jouée.

Figure I.2 : salle d'attente Çankaya résidence

Figure I.3 : la salle d'attente des urgences montréalaises.

3.2 **Système de gestion de file d'attente :**

Ce système est composé d'un ordinateur serveur, un distributeur de tickets, un grand écran afficheur, des baffles et des caméras. Apres la configuration du système et l'enregistrement des noms d'opérateurs, des guichets, des articles, des opérations ainsi que les relations entre les articles et les guichets, on gère les articles prioritaires.

3.3 **Définition d'un système :**

Un système est un ensemble d'éléments qui interagissent entre eux dans un environnement particulier en formant un tout. Il est donc le siège d'échanges et de relations plus ou moins complexes, ses caractéristiques permettent de l'identifier en tant qu'objet unique.

On le définit aussi comme étant la matérialisation d'une correspondance entre un ensemble de variables d'entrées et un ensemble de variables de sortie et sa réponse dépend de son état et de ses entrées, elle peut évoluer en fonction du temps (figure I.4) [Reix 02].

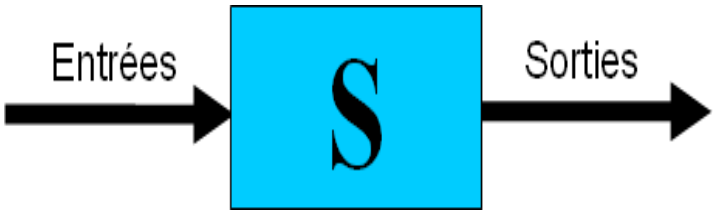

Figure I.4 : Système avec entrées/sorties

a) Objectif d'un système

Un système ne peut exister sans objectif, en effet l'ensemble des éléments qui interagissent dans le système sont organisés pour atteindre la raison d'être de tout système, qui n'est autre qu'un objectif bien déterminé.

b) Contrôle d'un système

Pour atteindre son objectif, un système doit être contrôlé, sans contrôle, la durée de vie d'un système est précaire.

Si les sorties s'écartent de l'objectif fixé, le contrôle agira sur les entrées ou sur la fonction de transformation du système ou bien sur les deux à la fois, pour minimiser cet écart.

La (figure I.5) [Reix 02] illustre le contrôle des sorties du système par rapport à son objectif.

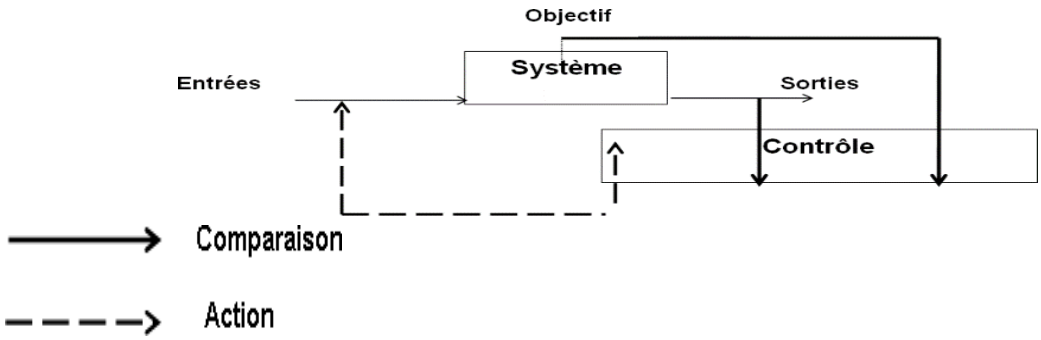

Figure I.5 : Contrôle des sorties d'un système

a) Différents types de système

On peut distinguer deux grandes familles de systèmes :

Les systèmes naturels : Ils sont utilisés pour décrire des phénomènes naturels et leurs propriétés, on cite : le système moléculaire, le système cellulaire, le système nerveux, le système immunitaire ….

Les systèmes artificiels : ils servent à décrire des concepts imaginés par l'activité humaine. Ils diffèrent selon les domaines : tels que domaine de technologie et des sciences appliquées. On a le système de télécommunication, pour les systèmes dédiés à la technologie de l'information. On a les systèmes d'exploitation, informatiques et les systèmes experts, etc.

3.4 **Système d'information**

Un système d'information (noté SI) représente un ensemble organisé d'éléments qui permet de gérer, regrouper, stocker, traiter, classifier, transporter et de diffuser de l'information sur un phénomène donné au sein d'une organisation.

En informatique et plus généralement dans le monde de l'entreprise, le terme système d'information (ou SI) possède les significations suivantes :

Un ensemble organisé de ressources (personnel, données, procédures, matériel, logiciel, …) permettant d'acquérir, de stocker, de structurer et de communiquer des informations sous forme de textes, images, sons, ou de données codées dans des organisations. Selon leur finalité principale, on distingue des systèmes d'information.

Système d'information appelé Information System (IS) : Ensemble constitué par la définition des processus des métiers et par celle des stocks et flux d'information éclairant ces processus.

Système informatique appelé [Reix 02] Information Technologie (IT) : Ensemble de moyens matériels et logiciels assurant le stockage, le traitement et le transport des données sous forme électronique.

Le système d'information coordonne grâce à l'information les activités de l'organisation et lui permet ainsi d'atteindre ses objectifs. Il est le véhicule de la communication dans l'organisation. De plus, le SI représente l'ensemble des ressources (les hommes, le matériel, les logiciels) organisées pour : collecter, stocker, traiter et communiquer les informations.

3.4.1. Objectif

Un système ne peut exister sans objectif [Reix 02] en effet, l'ensemble des éléments qui interagissent dans le système sont organisés pour atteindre un objectif bien déterminé, l'objectif est la raison d'être de tout système.

L'entreprise crée de la valeur en traitant de l'information, en particulier dans le cas des sociétés de service. Ainsi, l'information possède une valeur d'autant plus grande qu'elle contribue à l'atteinte des objectifs de l'organisation.

3.4.2. Fonctions

Peu importe les supports de cette information (informatiques ou télécommunications), un système d'information réalise toujours les quatre grandes fonctions suivantes :

- La collecte d'informations : il faut identifier le processus de « récolte » de l'information brute.
- La conservation : mémoriser les informations brutes ou résultats de traitement.
- La transformation : traitement, rapprochement, calcul et comparaison d'information.
- Diffusion : accès à la mémoire et échange entre les acteurs.

3.4.3. Les composants

On peut préciser le contenu d'un système d'information en décrivant ses trois familles de composants qui sont :

- La matière informationnelle

Elle comprend :

- La donnée : c’est la matière de base, par exemple une donnée chiffrée, le contenu des cellules d’un tableur ou bien un ensemble de données non traitées.
- L’information : c’est le résultat obtenu après traitement par le système informatique.
- La connaissance : est une information qui intègre un retour d'expérience comme l’avis d'un expert.
- La structure gère l’information, il est important de la structurer et de l’organiser avant de pouvoir l’exploiter dans un domaine précis et pour cela on utilise des méthodes ou techniques tels que : l’arborescence, les graphes, les classements, etc.
- Les traitements ce sont les transformations des données en information, plus généralement, c'est la transformation d'une matière informationnelle en une autre, on cite par exemple : le tri, la recherche, la projection, la transformation, etc.

3.4.4. Organisation

Une organisation est un ensemble d'éléments en interaction, regroupés au sein d'une structure régulée, ayant un système de communication pour faciliter la circulation de l'information, dans le but de répondre à des besoins et d'atteindre des objectifs déterminés.

L’organisation est un système complexe, ou sa mission est sa raison d’être, elle peut se modifier avec le temps en fonction des transactions entre le système et son environnement.

I.4 Conclusion

Ce chapitre nous a permis de bien comprendre et assimiler les notions théoriques principales relatives au système de gestion de salle d'attente ainsi que les différents types , son architecture et son constitution , l'étape suivante sera de faire une description générale sur les différents équipements qui rentre dans la composition de notre système en vue de réalisation.

Chapitre II

Description

II.1 Introduction

Dans le présent chapitre nous décrivons les différentes cartes électroniques utilisées pour réaliser les systèmes de gestion à taille réduite en particulier Raspberry Pi. Ses principes de fonctionnement ainsi que ses éléments constitutifs , étant donné que ces cartes sont nettement plus répondus qu'au reste des cartes à titre d'exemple Arduino. Suivi derrière par le langage de programmation utilisé ainsi que le système d'exploitation

II.2 Carte électronique

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'époxy ou de fibre de verre. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées.

- **Raspberry pi**

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et d'autres logiciels compatibles.

Figure II.1 : Le Raspberry Pi

2.1 Historique :

2.1.1 Conception :

En 2006, les premiers prototypes du Raspberry Pi sont développés sur des HDMI.[8] ,Ur3 500 £. [8] microcontrôleurs Atmel *ATmega 644*. Le schéma et le plan du circuit imprimé sont rendus publics. Cet ordinateur s'inspire *du BBC Micro* d'*Acorn Computer* (1981) et est destiné à encourager la jeunesse à la programmation. Le premier prototype ARM est intégré dans un boitier de la même taille qu'une clé USB avec un port USB d'un côté et un port HDMI de l'autre.

2.1.2 Prototype :

En août 2011, 50 cartes version Alpha sont construites, ces cartes étant identiques du point de vue fonctionnel au modèle B prévu mais elles sont plus grandes pour faciliter le débogage. Une démonstration montre la carte exécutant une distribution Debian avec un bureau LXDE, *Quaker 3* en 1080p et une vidéo en Full HD MPEG-4. En octobre 2011, une version de *RISC OS 5* tournant sur la carte est présentée. Après une année de développement la version grand public sera terminée en novembre 2012. En décembre 2011, 25 cartes modèle B ont été construites et testées. [14] Le design des cartes version Beta est le même que les cartes grand public. Une seule erreur a été découverte dans le design, certaines broches du CPU ne fonctionnaient pas correctement, l'erreur a été corrigée avant la première production.

La première semaine de l'année 2012, 10 premières cartes sont mises aux enchères

Sur eBay. L'une est achetée anonymement et donnée au Centre « for Comptine Historie », dans le *Suffolk* en Angleterre. Les 10 cartes qui représentaient un prix de 220 £ ont été vendues pour un total de 16 000 £.

2.1.3 Lancement:

La première série de 10 000 cartes est produite en Taiwan et en Chine. Les livraisons de la première série sont annoncées pour mars 2012 en raison de l'installation d'un mauvais connecteur Ethernet, mais la fondation annonce qu'elle s'attend à augmenter la production des futures séries sans difficulté.

Les ventes débutent le 29 février 2012 à 06:00 UTC. Au même moment est annoncé un modèle A à 256 MB de RAM au lieu des 128 MB prévus. Le site web de la fondation affiche

:*« Six ans après le début du projet, nous sommes presque à la fin de la première session de développement - cependant ce n'est que le début de l'histoire de Raspberry Pi.* ». [12]

En septembre 2012, 500 000 cartes ont été vendues. Le 6 avril 2012, la fondation annonce que le Raspberry Pi a obtenu la certification CE, demandée par les distributeurs pour pouvoir lancer la distribution auprès des premiers acheteurs. [13]

Au 22 mai 2012, 20 000 cartes ont été envoyées. En juillet 2012, 4 000 unités sont produites chaque jour.

En septembre 2012, la Fondation Raspberry Pi annonce une deuxième révision du modèle B. De plus, les futures séries seront fabriquées au Royaume-Uni, dans les usines Sony de *Pence*, au *Pays de Galles*. Il est estimé que 30 000 unités seraient produites par mois, créant 30 emplois. La fréquence du processeur est passé de 700 à 1 000 MHz.

En octobre 2013, c'est un million de Raspberry Pi qui a été produits au Royaume-Uni. Le deux millionième kit est envoyé entre le 24 et le 31 octobre.

En avril 2014, une nouvelle version est annoncée, elle divise la carte en deux parties : une partie calcul et une partie interface d'entrées-sorties. La partie calcul *Computer Module* comporte 512 Mo de mémoire vive et 4 Go de mémoire flash. La dimension de la carte est réduite au format SO-DIMM (environ 68 × 30 mm). La partie interface d'entrées-sorties *Computer Module IO Bocard* comporte des connecteurs HDMI et USB. Au mois de juin, trois millions de Raspberry Pi ont été vendus. [14]

2.1.4 Architecture matérielle :

Le Raspberry Pi possède un processeur ARM11 à 700 MHz. Il inclut 1, 2 ou 4 ports USB, un port RJ45 et 256 Mo de mémoire vive pour le modèle d'origine (512 Mo sur les dernières versions). Son circuit graphique BMC Vide core 4 en particulier permet de décoder des flux Blu-Ray full HD (1080p 30 images par seconde), d'émuler d’anciennes consoles et d'exécuter des jeux vidéo relativement récents[12] .

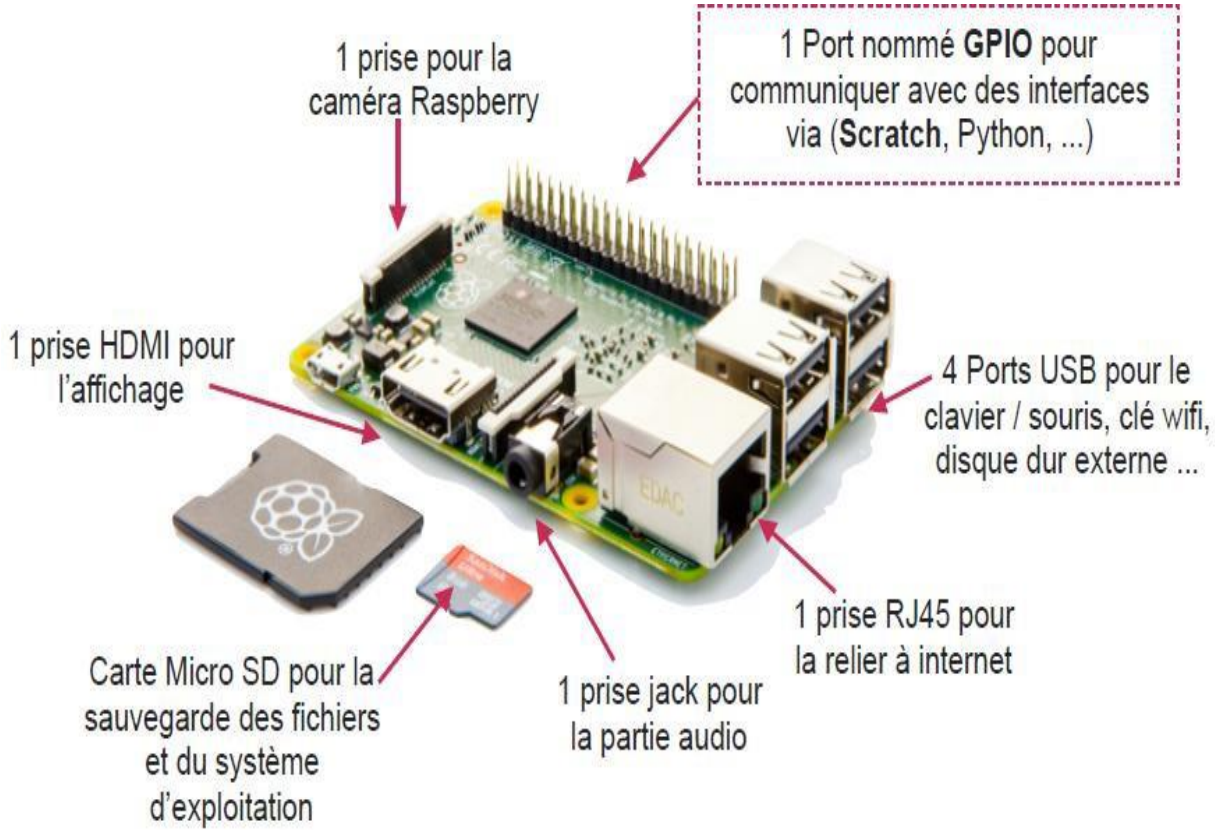

Figure II.2:Les composants standards d'un modèle Raspberry P i

2.2 Les différents modèles de Raspberry Pi

2.2.1 Raspberry Pi

Le produit de première génération a été lancé en 2012 et se décline en deux versions : A et B. Dans la deuxième année, les modèles A + et B + sont nés, ce qui nous a fait plaisir et a obtenu une bonne RAM. La différence entre ces modèles est la taille du GPIO, le nombre de ports USB et la présence d'Ethernet, la prise en charge de la carte MicroSD et la taille de la RAM.

Les nouvelles cartes sont très populaires dans le monde entier. Avec son processeur mono cœur cadencé à 700 MHz et 256 Mo de RAM (512 Mo pour le modèle B +), il a considérablement accru la curiosité de nombreux passionnés[11].

- **Modèle A :**

- Le Raspberry Pi A est équipé de 256 RAM et ne dispose pas de port Ethernet, et il est moins puissants.
- Sorties vidéo : Composite et HDMI 1 Sortie audio stéréo Jack3,5mm
- 1 Port USB2.0

Figure II.3 : Modèle *A*

- **Modèle A+ :**

- Plus petit que le Raspberry Pi A+, Lecteur de carte micro SD au lieu et place du lecteur SD et consommation électronique moindre.
- Processeur 64 bits quad core à 1,4 GHz Wifi bi-bande 2,4 GHz et 5 GHz Bluetooth 4.2/BLE.

Figure II.4 : Modèle A+

- **<u>Modèle B1 :</u>**

• Contrairement au modèle A, il possède deux ports USB, un port Ethernet et 512 RAM.
• Deux ports USB 2.0 au lieu de 512 Mo de RAM maximum et deux trous de montage.

Figure II.5 : Modèle B1

- **Modèle B1+ :**

- C’est la version finale du Raspberry Pi original,remplace le type B1
- GPIO 40 broches
- Ports USB2.0
- La consommation électrique est réduite de 3,5 W à3W

Figure II.6 : Modèle B1+

2.2.2 Raspberry Pi 2:

Trois ans plus tard, on a atteint la deuxième génération de produits en 2015. Raspberry Pi 2 sortira un nouveau modèle B, plus de single core, nous utiliserons un processeur quad-core 32 bits avec une fréquence d'horloge de 900Mhz, le tout pris en charge par 1 Go de RAM. Après avoir terminé le jeu, nous avons maintenant un véritable ordinateur avec des capacités multitâches et suffisamment de RAM pour les calculs. [11].

- **Modèle B :**

Il possède les mêmes dimensions et la même connectique que le modèle précédent.

Figure II.7 : Modèle B

2.2.3 Raspberry pi 3:

Enfin, fournissez-nous les produits de troisième génération au même prix que les produits de deuxième génération. Encore une fois, il s'agit d'une mise à niveau du processeur, nous utilisons 4 cœurs, mais nous avons mis à niveau vers 64 bits, avec un débit de 1,2 GHz. Plus précisément, notre puissance de calcul pure a augmenté de + 50% par rapport au produit de deuxième génération. Cependant, la vraie différence n'est pas la puissance, mais les «accessoires». Ce nouveau Raspberry dispose du Wi-Fi et du Bluetooth [11].

- **Modèle B3 :**

Le Raspberry Pi 3 Model B est plus efficace que le Raspberry Pi 2. Il est désormais équipé nativement du Wifi b / g / n et du Bluetooth 4.1.et processeur Broadcom BCM2837 64 bits, quatre cœurs ARM Cortex-A53 1,2 GHz, puces Wi-Fi 802.11n et Bluetooth 4.1 intégrées

- *Figure II.8 : Modèle B3*

- **Modèle A3+ :**

Le Pi A3 + a un processeur plus puissant que B + d'une part et d'autre part il n'est pas aussi bon que B + en matière de RAM, il y a donc moins de slots USB A 2.0. Par Broadcom BCM2837B0. ARM Cortex-A53 quadri-cœur 1,4 GHz

Figure II.9 : Modèle A3+

- **Modèle B3+ :**

- RPi 3 B + est une amélioration de type B. Le processeur Broadcom BCM2837B0 est 200 Mhz plus élevé que son prédécesseur.
- Broadcom BCM2837B0 Wifi double bande 802.11ac version 4.2. 1,4 GHz Quatre cœurs ARM Cortex- A53.

Figure II.10 : Modèle B3+

2.2.4 Raspberry pi Zéro :

Le 26 novembre 2015, la Fondation Raspberry Pi a annoncé le lancement du Raspberry Pi Zéro. Utilise les spécifications du modèle A / B et la fréquence d'horloge du processeur est de 1 GHz au lieu de 700 MHz. En revanche, il est plus petit et a le moins de connexions. Il en existe trois types: 0, 0 W et 0 WH [11].

- **Modèle zéro :**

Le Raspberry Pi Zéro fait la moitié de la taille du modèle A + et est deux fois plus pratique et élégant.

Figure II.11 : Modèle zéro

- **Modèle zéro W :**

Raspberry Pi Zéro W : la série Pi Zéro avec d'autres connexions Bluetooth et LAN sans fil.

Processeur :1GHz

RAM :512Mo

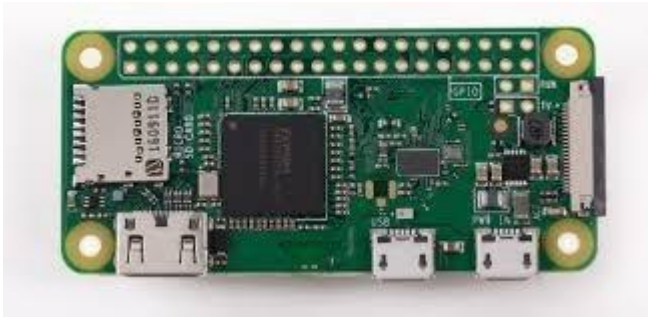

Figure II.12 : Modèle zéro W

- **Modèle zéro WH :**

La même carte que le Raspberry Pi Zéro W, mais avec un connecteur GPIO 40 broches soudé.

SPI+I2C

Caméra+Micro SD

Figure II.13 : Modèle zéro WH

2.2.5 Raspberry Pi 4 :

- Le 24 juin 2019, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 4 avec un modèle B est de :
- Processeur 1.5GHz quad-core 64-bit ARM Cortex-A72.• 1GB, 2GB, ou 4GB de mémoire SD RAM LP DDR4.
- Ethernet Gigabit.
- WifiDual-band802.11ac.• Bluetooth5.0.
- Deux ports USB 3.0 et deuxportsUSB2.0
- Support double ecran.
- GPU VideoCore.

Figure II.14 : Modèle RPi 4

Pour notre projet, nous avons choisi le Raspberry Pi B3

2.3 Caractéristiques du Raspberry Pi utilisé :

Le RaspberryPi est un nano-ordinateur de la taille grosso modo d'une carte de crédit. Ce qui est impressionnant, ce sont ses caractéristiques. Sur un petit bout de carte électronique, il y a un:

—Processeur ARM Cortex A-53 quatre cœurs de 64-bit cadencé à 1,2 Ghz

—1 Go de RAM (SDRAM)

—Un contrôleur vidéo Broadcom Vide Coré IV

—4 ports USB 2.0

—Un port HDMI

—Un connecteur Jack 3,5 mm / sortie composite

—Une prise Ethernet 10 / 100 Mbps

—WIFI802.11n

—Bluetooth4.1

—Bluetooth Low Energy (BLE)

—40 pins GPIO

—Une interface de caméra (CSI)

—Une interface d'affichage (DSI)

—Un emplacement de carte micro SD

—Alimentation 5V par micro USB

La figure qui suit inventorie les différentes parties d'un Raspberry Pi 3 modèle B,

2.3.1 Accessories du Raspberry Pi

a) Alimentation

Le Raspberry Pi s'alimente sous tension unique de 5 volts, tension sur laquelle il peut consommer jusqu'à1.8A selon les taches qu'il exécute. Cette alimentation doit être normalement fournie via le connecteur micro USB place dans un angle de la carte.

En effet, l'utilisation d'un chargeur pour Smartphone équipé d'un micro USB qui délivre une tension de 5 volts avec au minimum 1.8A est suffisant pour alimenter notre Raspberry Pi.

b) Carte microSD

Pour héberger notre système d'exploitation(Raspbian) nécessaire au bon fonctionnement notre Raspberry Pi, et tous les fichiers nous auront besoin d'espace de stockage c'est pour cela qu'une micro SD est requise.

La qualité de cette carte, et tout particulièrement sa vitesse, va influencer la vitesse d'écriture des fichiers. Il convient donc de bien la choisir .

2.3.2 Que peut-on faire avec un Raspberry Pi ?

On peut pratiquement faire avec un Raspberry Pi tout ce que l'on peut faire avec un ordinateur de bureau sous Linux, à quelques exceptions prés. Le Raspberry Pi utilise une carte SD à la place d'un disque dur, bien que l'on puisse le brancher à un disque dur USB.

On peut modifier des documents bureautiques, surfer sur internet, jouer à des jeux, etc.

Le Raspberry Pi est un appareil extrêmement souple qu'on peut utiliser dans toutes sortes de situations, que ce soit pour remplacer un ordinateur de bureau, profiter d'un media center ou contrôler un système à l'aide d'un ordinateur embarqué.

II.3 Linux

Un ordinateur standard fonctionne avec un système d'exploitation comme Windows, OS X ou Linux. C'est ce système qui démarre quand vous allumez votre machine et qui fournit à vos logiciels un accès aux fonctionnalités matérielles de votre ordinateur. Par exemple, si vous codez une application qui utilise Internet, vous pouvez, pour ce faire, utiliser les fonctions fournies par le système d'exploitation. Vous n'avez pas besoin de comprendre et d'écrire du code pour chaque modèle d'interface Ethernet ou Wi-Fi disponible.
Comme tous les autres ordinateurs, le Raspberry Pi utilise un système d'exploitation.
Celui fourni par défaut est une version de GNU/Linux appelée Raspbian qui est idéal pour l'esprit de partage qui règne au sein de la communauté Raspberry Pi car il est libre et open source 5. Ce logiciel a été écrit dans le cadre d'un projet communautaire pour ceux qui recherchent une alternative au monopole de Microsoft Windows et d'Apple OS X.
Linux est un système d'exploitation Open Source. C'est un système d'exploitation complet, basé sur les mêmes concepts robuste d'UNIX qui est apparu au début de l'informatique. Il a de nombreux partisans fidèles et serviables et s'est transformé en un système d'exploitation puissant et facile à utiliser.
Bien que le système d'exploitation s'appelle Linux, plusieurs distribution (appelées aussi distors) Linux différentes ont été produites. Celles-ci comprennent le même système d'exploitation de base, mais sont livrées avec des applications ou de systèmes de fenêtrage différents.
Bien qu'il existe de nombreuses distributions, C'est la Rasbian qui est recommandée par la fondation Rasberry Pi.[4]

3.1 NOOBS :

Trouver son système d'exploitation grâce à NOOBS lorsque l'on débute avec son Raspberry Pi, on se retrouve rapidement perdu lorsqu'il faut choisir un système d'exploitation sachant que sous Linux, il existe de nombreuses distributions différentes. Heureusement, le logiciel « NOOBS » est là pour nous.

3.2 Définition :

(« News Out Of the Box Software ») est un logiciel prévu pour tester facilement les systèmes d'exploitations classiques que l'on retrouve couramment sur Raspberry Pi. C'est un assistant d'installation développé par Raspberry Pi Foundation, destiné à simplifier l'installation des utilisateurs débutants dans le secteur Linux .

- Raspbian : la célèbre distribution classique basée du Debian
- Pidora : une distribution basée sur Fedora
- Open Elec : pour transformer le Pi en centre multimédia
- RaspBMC : idem qu'Open Elec
- et d'autres comme Arch OS ou Risc OS...

Grâce à ce logiciel, il est possible d'installer plusieurs systèmes d'exploitation sur une seule et même carte SD : ainsi, nous pouvons facilement trouver celui qui est le plus adapté à nos besoins sans avoir à télécharger les différentes images nécessaires.

3.3 Raspbian

Raspbian est un système d'exploitation GNU/Linux spécialement conçu et optimisé pour le Raspberry Pi. [7]

3.3.1 Caractéristiques

Raspbian est un mot-valise 6 formé par la fusion des mots "Raspberry Pi" et "Debian".

Figure II.15 : RaspberryPi+Debian=Raspbian

II.4 Python :

Nous verrons dans le chapitre suivant qu'il est possible de relier ces lignes à une multitude de composants externes, capteurs, actionneurs ...etc., mais pour faire fonctionner tous ces éléments, il faudra écrire des programmes spécifiques qui seront écrit en Python.
Pourquoi Python et pas un autre langage comme Java, C, C++, Basic, tous plus connus au moins du grand public ?

Pour au moins quatre raisons :
— Python est un langage facile à apprendre par qui n'a jamais programmé. La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. A fonctionnalités égales, un programme Python (abondamment commenté et présenté selon les canons standards) est souvent de 3 à 5 fois plus court qu'un programme C ou C++ (ou même Java) équivalent, ce qui représente en général un temps de développement

de 5 à 10 fois plus court et une facilité de maintenance largement accrue.
— Le langage de programmation officielle qui a été pris en charge comme langue d'apprentissage éducatif sur le Raspberry Pi (le langage de prédilection á l'origine du Pi de Raspberry Pi) est Python, Python est d'ailleurs installé par défaut dans Raspbian.
— Python est un langage de programmation interprété et qu'il est donc très facile d'essayer immédiatement une ou plusieurs instructions dont on n'est pas sur du comportement.
— De nombreuses bibliothèques sont disponibles sur internet pour piloter les lignes d'entrées/ sorties du Raspberry Pi en Python.

Figure II.16 : Logo officiel de Python

4.1 Programmation des entrées / sorties avec python

Les broches marquées GPIO peuvent être utilisées comme broches d'entrée / sortie. En d'autres termes, n'importe quelle broche peut être programmée comme une entrée ou une sortie.

Dans ce sens on va utiliser plusieurs langages de programmation capable de contrôler ces broches comme le C, Java, ... mais dans notre projet on a opté pour le python pour contrôler ces broches.

Le module GPIO est installé par défaut sur les versions les plus récentes de la distribution Raspbian Linux. Mais pour les versions plus anciennes, on doit probablement l'installer et effectuer une mise à jour. Pour cela on exécute la commande suivante :

"*sudo apt-get install python-rpi.gpio*" Ensuite on exécute la commande suivante pour la mise à jour :

"*sudo apt-get update*"

II.5 Conclusion

Dans ce chapitre nous avons passé en revue le raspberry pi et ses variétés d'applications et d'utilisations nous avons également parlé du système d'exploitation linux et enfin nous avons parlé du langage de programmation python et comment nous l'avons appliqué dans notre projet que nous allons approfondir dans le prochain chapitre.

Chapitre III

Programmation

III.1 Introduction :

Dans ce chapitre, nous utiliserons notre langage de programmation (python) pour créer notre programme de projet composé de trois fichiers principaux : Interface graphique, client, serveur

III.2 Interface graphique :

III.2.1 Affichage d'une fenêtre simple :

Pour afficher une fenêtre simple on installe le module tkinter (Tkinter est la bibliothèque d'interface utilisateur graphique (GUI) standard pour Python Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl (Tool Command Langage), et on l'importe dans le programme.

Après, on va créer une fenêtre avec une hauteur de 1350 et une largeur de 756, son titre est Système de salle d'attente, et on utilise «window.mainloop()» pour l'exécuter.

```
zeggai@zeggai-VirtualBox:~$ sudo apt-get install python3-tk
```

Figure III.1 : commande d'installation de tkinter

```
1  from tkinter import *
2
3  window=Tk()
4  window.geometry('1350x756')
5  window.title("Système de salle d'attente")
6  window.mainloop()
7
```

Figure III.2 : Programme d'une fenêtre simple

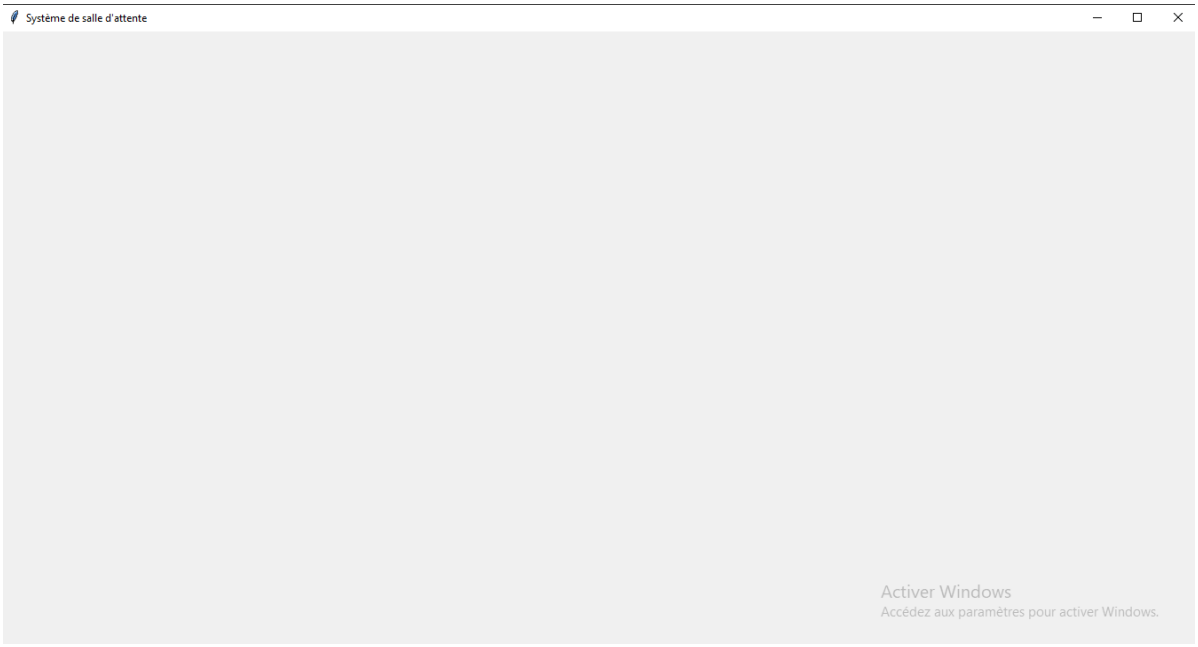

Figure III.3 : Affichage d'une fenêtre simple

III.2.2 Insertion d'image :

On crée un espace rectangulaire avec une hauteur de 1350 et une largeur de 756 (1350x756) en utilisant Canvas (canvas est une zone rectangulaire destinée à dessiner des images ou d'autres mises en page complexes. On peut placer des graphiques, du texte, des widgets..)

Ensuite on va insérer une image fond d'écran de type png en tilisant «photo=PhotoImage(..)» et un logo en utilisant «photo1=PhotoImage(..)» .

Pour afficher une image sur un canvas, on utilise l'instruction:

-Canvas.create_image(x, y, option, ...) : retourne l'identifiant numérique de l'item image créée sur le canevas appelant.

On place le tout dans l'espace créé on utilisant «C.pack()» .

```
from tkinter import *

window=Tk()
window.geometry('1350x756')
window.title("Système de salle d'attente")

C = Canvas(window,width=1350, height=756,bd=0,highlightthickness=0)

photo = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bg.png")
C.create_image(0,0,anchor=NW,image=photo)

photo1 = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bb.png")
C.create_image(589,360,anchor=NW,image=photo1)

C.pack()

window.mainloop()

```

Figure III.4 : Programme canvas

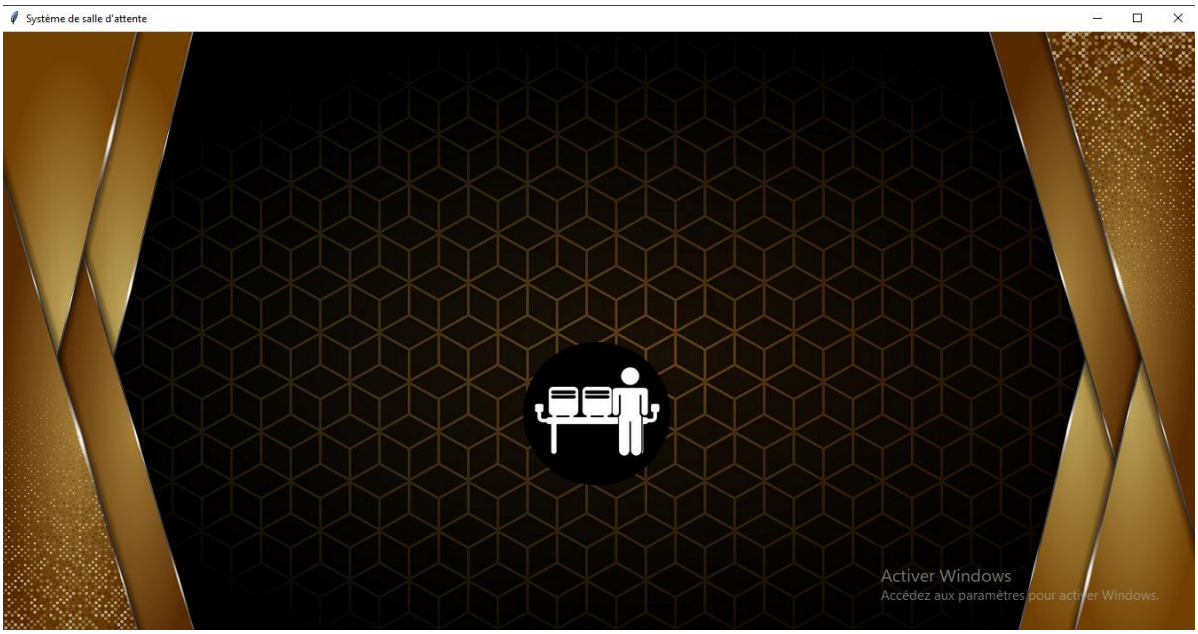

Figure III.5 : Affichage d'image et de logo

III.2.3 Affichage des textes :

A) Texte normal :

On peut afficher une ou plusieurs lignes de texte sur un canvas « C » en créant un objet texte.

a = C.create_text (x, y, option, ...) .Cela renvoie l'ID numérique de l'objet texte sur le canvas C. Les options incluent dans notre programme :

fill : couleur du texte.

font : utiliser cette option pour changer la police de caractères.

text : le texte à afficher sous la forme d'une chaîne de caractères.

```python
from tkinter import *

window=Tk()
window.geometry('1350x756')
window.title("Système de salle d'attente")

C = Canvas(window,width=1350, height=756,bd=0,highlightthickness=0)

photo = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bg.png")
C.create_image(0,0,anchor=NW,image=photo)

photo1 = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bb.png")
C.create_image(589,360,anchor=NW,image=photo1)

a=C.create_text(585,100,text='Bienvenue',fill = "white",font=("Times",50,"bold"))
b=C.create_text(280,300,text="Guichet",fill = "white",font=("Times",45,"bold"))
c=C.create_text(880,300,text='Ticket',fill = "white",font=("Times",45,"bold"))
gu=C.create_text(400,450,text="000",fill = "gold",font=("Times",50,"bold"))
ti=C.create_text(920,450,text="000",fill = "gold",font=("Times",50,"bold"))

C.pack()

window.mainloop()
```

Figure III.6 : Programme des textes normal

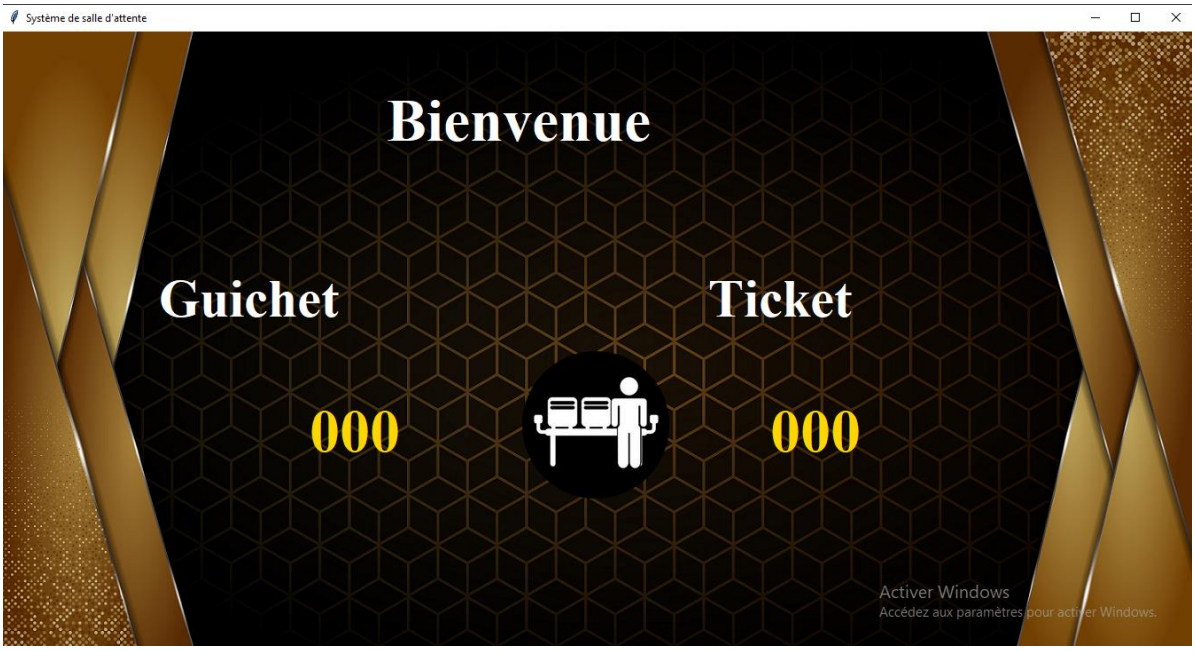

Figure III.7 : Affichage des textes normal

B) Texte en arabe :

L'écriture arabe est très spéciale avec deux caractéristiques essentielles :

Il s'écrit de droite à gauche.

Les personnages changent de forme en fonction de leurs personnages environnants.

Donc nous avons deux problèmes, d'une part, les caractères sont sous forme isolée, ce qui signifie que chaque caractère est rendu quel que soit son environnement, et d'autre part, le texte est écrit de gauche à droite.

Pour cela, on va installer deux bibliothèques, «bidi.algorithm» et «arabic_reshaper».

```
zeggai@zeggai-VirtualBox:~$ pip install arabic-reshaper
```

Figure III.8 : Commande pour installer arabic_reshaper

```
zeggai@zeggai-VirtualBox:~$ pip install python3-bidi
```

Figure III.9 : commande pour installer bidi.algorithm

Et on les importe sur le programme.

On utilise « «arabic_reshaper.reshape('..') » pour remodeler les lettre et « get_display(..) » pour lier les lettres séparées .

```
1  from tkinter import *
2  #import arabic_reshaper
3  #from bidi.algorithm import *
4
5  window=Tk()
6  window.geometry('1350x756')
7  window.title("Système de salle d'attente")
8
9  C = Canvas(window,width=1350, height=756,bd=0,highlightthickness=0)
10
11 photo = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bg.png")
12 C.create_image(0,0,anchor=NW,image=photo)
13
14 photo1 = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bb.png")
15 C.create_image(589,360,anchor=NW,image=photo1)
16
17 a=C.create_text(585,100,text='Bienvenue',fill = "white",font=("Times",50,"bold"))
18 b=C.create_text(280,300,text="Guichet",fill = "white",font=("Times",45,"bold"))
19 c=C.create_text(880,300,text='Ticket',fill = "white",font=("Times",45,"bold"))
20 gu=C.create_text(400,450,text="000",fill = "gold",font=("Times",50,"bold"))
21 ti=C.create_text(920,450,text="000",fill = "gold",font=("Times",50,"bold"))
```

Figure III.10 : Programme des textes en arabe 1

```
#a1 =arabic_reshaper.reshape("مرحبا")
a1="مرحبا"
#a1 = get_display(a1)
e=C.create_text(830,100,text=a1,fill = "white",font=("Times",50,"bold"))
#a2 =arabic_reshaper.reshape("الشباك")
a2="الشباك"
#a2 = get_display(a2)
f=C.create_text(485,300,text=a2,fill = "white",font=("Times",50,"bold"))
#a3 =arabic_reshaper.reshape("التذكرة")
a3="التذكرة"
#a3 = get_display(a3)
g=C.create_text(1070,300,text=a3,fill = "white",font=("Times",50,"bold"))

C.pack()

window.mainloop()
```

Figure III.11 : Programme des textes en arabe 2

Figure III.12 : Affichage des textes en arabe

C) Heur et date :

- **Affichage de date :**

On importe le module « datetime » .

«d = date.today() » :permet d'obtenir la date d'aujourd'hui.

« C.create_text (x, y, option, ...) » :cela renvoie l'ID numérique de l'objet texte sur le canvas C. Les options incluent dans notre programme :

fill : couleur du texte.

font : on utilise cette option pour changer la police de caractères.

text : le texte a affiché la date sous la forme d'une chaîne de caractères.

- **Affichage d'heure :**

On importe le module time.

Ensuite on va créer une fonction « display time () » .

Dans cette fonction on a :

-une valeur global « clock_label ».

-« strftime(..) » : pour obtenir le temps .

-« window.after(1000.display_time) » : veut dire que la fonction se répète chaque 1000 ms .

-« C.itemconfig(clock_label,text=current_time) » : pour configurer le changement de texte de la valeur clock_label .

On ajoute alors la fonction « C.create_text (x, y, option, ...) » et display_time() pour exécuter la fonction .

```
from tkinter import *
#import arabic_reshaper
#from bidi.algorithm import *
from datetime import *
from time import *

window=Tk()
window.geometry('1350x756')
window.title("Système de salle d'attente")

C = Canvas(window,width=1350, height=756,bd=0,highlightthickness=0)

photo = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bg.png")
C.create_image(0,0,anchor=NW,image=photo)

photo1 = PhotoImage(file=r"C:\Users\lenovo\Desktop\Nouveau dossier\\bb.png")
C.create_image(589,360,anchor=NW,image=photo1)

a=C.create_text(585,100,text='Bienvenue',fill = "white",font=("Times",50,"bold"))
b=C.create_text(280,300,text="Guichet",fill = "white",font=("Times",45,"bold"))
c=C.create_text(880,300,text='Ticket',fill = "white",font=("Times",45,"bold"))
gu=C.create_text(400,450,text="000",fill = "gold",font=("Times",50,"bold"))
ti=C.create_text(920,450,text="000",fill = "gold",font=("Times",50,"bold"))

d=date.today()
d=C.create_text(300,30,text=d,fill = "white",font=("Times",25,"bold"))
```

Figure III.13 : Programme de date

```python
def display_time():
    global clock_label
    current_time = strftime('%H:%M:%S')
    window.after(1000,display_time)
    C.itemconfig(clock_label,text=current_time)

clock_label=C.create_text(1045,30,text="",fill = "white",font=("Times",25,"bold"))

#a1 =arabic_reshaper.reshape("مرحبا")
a1="مرحبا"
#a1 = get_display(a1)
e=C.create_text(830,100,text=a1,fill = "white",font=("Times",50,"bold"))
#a2 =arabic_reshaper.reshape("الشباك")
a2="الشباك"
#a2 = get_display(a2)
f=C.create_text(485,300,text=a2,fill = "white",font=("Times",50,"bold"))
#a3 =arabic_reshaper.reshape("التذكرة")
a3="التذكرة"
#a3 = get_display(a3)
g=C.create_text(1070,300,text=a3,fill = "white",font=("Times",50,"bold"))

display_time()

C.pack()

window.mainloop()
```

Figure III.14 : Programme d'heur

Figure III.15 : Affichage d'heur et de date

-On ajoute une fonction maj(ss) ,sur cette fonction, on a des valeurs globales « gu ,ti, gui, tic », un dossier « paramSD + la valeur string ss » , on a aussi « os.path.getsize(sss) » pour obtenir la taille de fichier.

Si cette taille et plus grande ou égale a 1, on met :

« open(sss,w,utf8) » pour ouvrir le fichier sss dans le but d'y écrire .

« tic=tic+1 », parce que le nombre de tickets augmente chaque fois a 1.

« C.itemconfig(..) » pour configurer le changement des tickets et des guichets dans le canvas.

```
20  def maj(ss):
21      global gu,ti,gui,tic
22
23      sss=r"C:\Users\lenovo\Desktop\Nouveau dossier\paramSD"+str(ss)+".aaa"
24
25      nn=os.path.getsize(sss)
26      if nn>=1:
27          f = open(sss,'w',encoding="utf8")
28
29          tic=tic+1
30          C.itemconfigure(gu,text=str(ss))
31          C.itemconfigure(ti,text=str(tic))
```

Figure III.16 : Fonction maj

Supposons qu'on a 10 guichets donc on écrit : maj(1),maj(2)…..maj(10).

```
49  maj(1)
50  maj(2)
51  maj(3)
52  maj(4)
53  maj(5)
54  maj(6)
55  maj(7)
56  maj(8)
57  maj(9)
58  maj(10)
59
```

Figure III.17 : Nombre de guichets

III.3 Application client :

On importe 2 modules : tkinter et socket.

La programmation de socket est un moyen de connecter deux nœuds sur un réseau pour communiquer entre eux. Un socket (nœud) écoute sur un port particulier à une adresse IP, tandis qu'un autre socket se connecte à l'autre pour former une connexion. Le serveur forme le socket d'écoute pendant que le client contacte le serveur.

-On commence par fixer l'adresse ip .

-Ensuite, on met une fonction comm_1 , sur cette fonction, on va écrire un socket simple qui a deux paramètre , « AF_INET » pour le type d'adresse ipv4 et « SOCK_STREAM » veut dire c'est un TCP socket .

TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission.

-On va connecter le socket créé a l'adresse ip écrite + le numéro du port utilisé.

-Après on utilise la méthode « ss=entree.get() » pour obtenir une entrée et on envoi le numéro du guichet «ss» . Nous utilisons la méthode « s.send(ss.encode()) ».

.

```
1   from tkinter import *
2   import socket
3
4   hote = "192.168.1.114"
5
6   def comm_1():
7
8       s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9       s.connect((hote, 1111))
10      ss=entree.get()
11
12      s.send(ss.encode())
13
```

Figure III.18 : Programme de l'application client 1

-Maintenant, on va créer une fenêtre, dans cette dernière, on a une entrée avec des options width , justify , et fonts .

On insert le numéro 1 par défaut, et en pack l'entrée en haut de la fenêtre avec padx et pady à la valeur 3, on a aussi un button b1, pour la commande comm_1, avec les options fonts, text, heigh, width.

-Ensuite en va créer un canvas dans la fenêtre avec les options height , width , bg et on le pack en haut avec un padx et pady=5.

-fen.mainloop () pour l'exécution.

```
14
15  ##### Programme principal : ############
16  fen = Tk()
17  entree = Entry(fen,width=4,justify="center",font=('Times', 20, 'bold'))
18  entree.insert(0,"1")
19  entree.pack(side =TOP, padx =3, pady =3)
20
21  can = Canvas(fen, width =400, height =300, bg ='ivory')
22  can.pack(side =TOP, padx =5, pady =5)
23
24  b1 = Button(fen,font=('Times', 20, 'bold'), text ='suivant',heigh=1,width=10, command =comm_1)
25  b1.pack(side =BOTTOM, padx =3, pady =3)
26
27  fen.mainloop()
```

Figure III.19 : Programme de l'application client 2

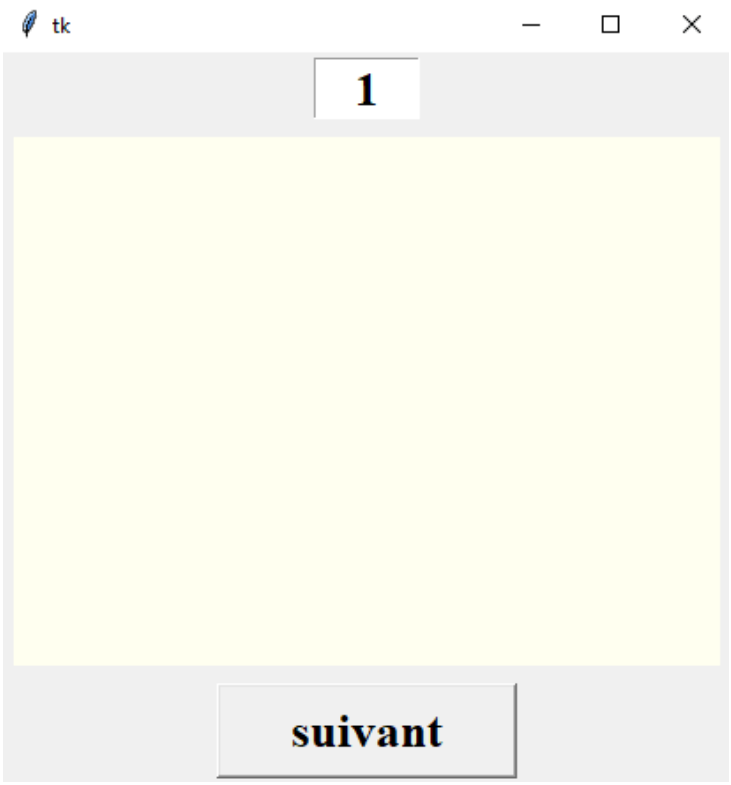

Figure III.20 : Application Client

III.4 Application serveur :

On va importer deux modules : socket et threading, et en écrit spécifie l'adresse IP.

(Le module threading est utilisé pour créer, contrôler et gérer des threads en python).

On va mettre une nouvelle classe « ClientThread », (la classe est comme un constructeur d'objets, ou un plan pour créer des objets), dans cette classe on a deux fonctions :

- La fonction __INIT__ : dans cette fonction on écrit « threading.Thread.__init__ (self) », et« self.ip=ip … . »

La méthode __init__ peut être appelée lorsqu'un objet est créé à partir de la classe, et un accès est requis pour initialiser les attributs de la classe.

Le mot-clé self in en Python est utilisé pour toutes les instances d'une classe. En utilisant le mot-clé self, on peut facilement accéder à toutes les instances définies au sein d'une classe, y compris ses méthodes et attributs.

Si on clique sur le bouton « suivant » le programme affiche « [+] Nouveau thread pour 192.168.1.114 »

-La fonction run : on écrit une liste tfi et on utilise la méthode « .recv() »pour recevoir le socket du client et on le décode. On utilise la méthode split() pour diviser la chaîne reçue en une liste, et on affiche la valeur de cette chaine en utilisant « print ».

On écrit aussi le dossier paramSD + la liste tfi a 0 et nous l'ouvrons en utilisant « open ».

La liste tfi a 0 égale a la chaine reçus « r »

On écrit les éléments de la liste tfi dans le fichier avec la fonction « writelines » et on le ferme en utilisons « close ».

```python
import socket
import threading

hote="192.168.1.114"

class ClientThread(threading.Thread):
    def __init__(self, ip, port, clientsocket):
        threading.Thread.__init__(self)
        self.ip = ip
        self.port = port
        self.clientsocket = clientsocket
        print("[+] Nouveau thread pour %s %s" % (self.ip, self.port, ))
    def run(self):
        tfi=[]
        r = self.clientsocket.recv(2048)
        r=r.decode()
        tfi=r.split(" ")
        print(r)
        #print("Client déconnecté...")
        ss=r"C:\Users\lenovo\Desktop\Nouveau dossier\paramSD"+tfi[0]+".aaa"
        f = open(ss,'w',encoding="utf8")
        tfi[0]=r
        f.writelines(tfi)
        f.close()
```

Figure III.21 : Programme de serveur 1

-On va créer un nouveau socket pour le serveur. Pour ceci on écrit « setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADD,1) »

-La fonction setsockopt() fournit à un programme d'application les moyens de contrôler le comportement des sockets. Un programme d'application peut utiliser setsockopt() pour allouer de l'espace tampon, contrôler les délais d'attente ou autoriser les diffusions de données de socket.

-SOL_SOCKET est la couche socket elle-même. Elle est utilisée pour les options qui sont indépendantes du protocole.

-SO_REUSEADDR permet à un socket de se lier de force à un port utilisé par un autre socket.

Et on lie le socket à l'adresse IP + le numéro du port.

```python
tcpsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcpsock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
tcpsock.bind((hote,1111))
```

Figure III.22 : Programme serveur 2

On écrit maintenant une boucle while, Si vrai, donc le socket écoute « .listen »

Le programme affiche En écoute … en utilisant « print ».

Ensuite le socket de serveur accepte le socket de client.

Puis, on crée un nouveau thread qui s'appelle « ClientThread », avec l'adresse IP , le numéro de port + le socket de client.

Maintenant, le thread démarre « .start() ».

```
31  while True:
32      tcpsock.listen(10)
33      print( "En écoute...")
34      (clientsocket, (ip, port)) = tcpsock.accept()
35      newthread = ClientThread(ip, port, clientsocket)
36      newthread.start()
```

Figure III.23 : Programme serveur 3

-Finalement, on exécute le serveur puis le client et après l'interface graphique.

On suppose que le numéro du guichet est égale à 1, si on clique sur suivant, on obtient :

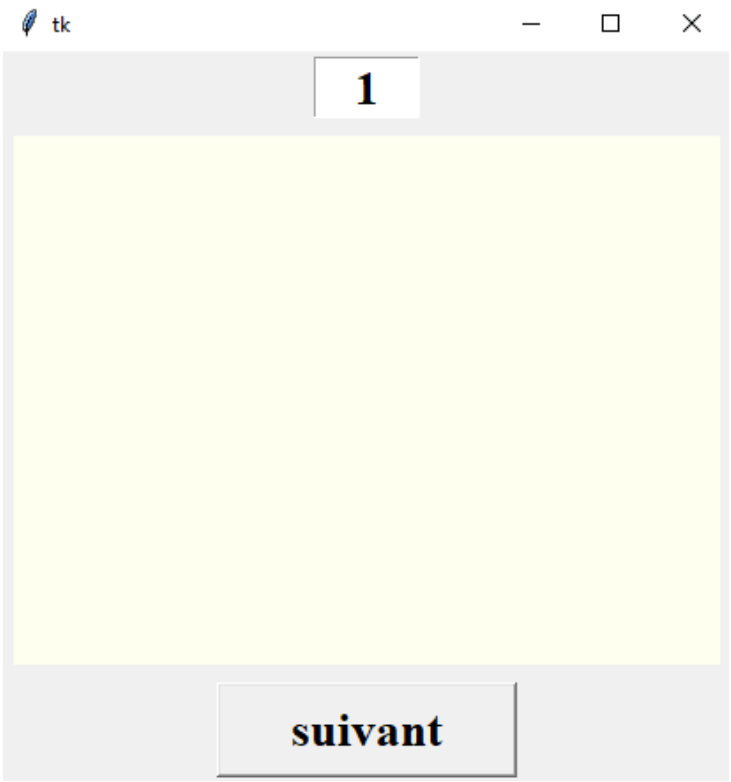

Figure III.24 : Application Client

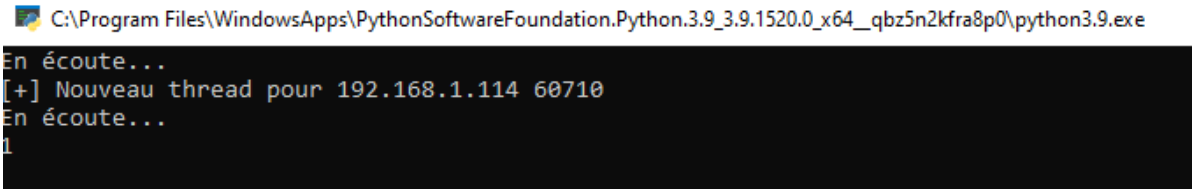

Figure III.25 : Exécution de serveur

Figure III.26 : Affichage final

III.5 Conclusion

Dans ce chapitre, nous avons appris à programmer notre salle d'attente et nous l'avons fait en utilisant le langage de programmation python pour créer et lancer notre projet, ce qui a rendu le processus d'attente plus facile et utilisé de manière très simple.

Conclusion générale

En conclusion, ce mémoire nous a permis d'atteindre des nouveaux objectifs.

En effet, ce projet nous a permis de comprendre le fonctionnement d'un système de gestion d'une salle d'attente.

Ensuite, nous avons appris beaucoup sur les caractéristiques d'un rasbperry pi et ses usages.

Puis, nous avons utilisé quelques commandes de base de système Linux.

Et enfin, le plus important, c'est la maitrise des bases nécessaires, des sockets et de la bibliothèque d'interface graphique du langage python.

Bibliographie

https://www.companeo.com/securite-electronique/FAQ/definition-serrure-autonome-electronique [1]

https://www.companeo.com/securite-electronique/FAQ/-technologies-de-lecteur-badge-magnetique-rfid-ou-sans-contact [2]

https://www.companeo.com/securite-electronique/FAQ/definition-controle-acces-biometrique [3]

https://fr.wikipedia.org/wiki/Raspberry_Pi 5555[4]

https://www.raspberrypi-france.fr/[5]

https://www.raspberrypi-france.fr/guide/[6]

https://www.google.com/search?client=opera&q=usb+rfid+reader&sourceid=opera&ie=UTF-8&oe=UTF-8 [7]

https://pubs.opengroup.org/onlinepubs/009695399/functions/setsockopt.html[8]

https://stackoverflow.com/questions/21515946/what-is-sol-socket-used-for[9]

https://docs.microsoft.com/en-us/windows/win32/winsock/using-so-reuseaddr-and-so-exclusiveaddruse[10]

https://www.w3schools.com/python/python_classes.asp#:~:text=Previous%20Next%20%E2%9D%AF-,Python%20Classes%2FObjects,%22blueprint%22%20for%20creating%20objects . [11]

https://www.w3schools.com/python/ref_string_split.asp[12]

https://www.edureka.co/blog/init-in-python/[13]

https://www.geeksforgeeks.org/socket-programming-python/[14]

https://www.w3schools.com/python/ref_file_writelines.asp[15]

https://www.commentcamarche.net/contents/538-le-protocole-tcp[16]

https://mpcabd.xyz/python-arabic-text-reshaper/[17]

https://docs.huihoo.com/tkinter/tkinter-reference-a-gui-for-python/create_text.html[18]

https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/create_text.html[19]

https://www.larousse.fr/dictionnaires/francais/syst%C3%A8me/76262[Reix 01]

الملخص

يتحدث مشروعنا عن المشكلات التي واجهها الأشخاص قبل وجود نظام غرفة الانتظار ويقدم أيضًا طريقة لفهم كيفية عمل نظام غرفة الانتظار.

بعد ذلك، تعرفنا على ميزات Raspberry Pi وأنواعها والعديد من الاستخدامات ونظام التشغيل Raspbian.

بعد ذلك، استخدمنا بعض أوامر Linux الأساسية.

في الختام، تمكنا من برمجة مشروعنا من خلال Python باستخدام مكتبات TKinter و Socket وبعض الأساسيات

Abstract

Our project talks about the problems that people faced before the waiting room system existed, and also presents a way to understand the working of the waiting room system.

Next, we learned about the features of Raspberry Pi, its types, its many uses, and its Raspbian operating system.

Then we used some basic Linux commands.

In conclusion, we were able to program our project through Python using the TKinter and Socket libraries and some of the basics

Résumé

Notre projet décrit les problèmes auxquels les gens étaient confrontés avant l'existence des systèmes de gestion de la salle d'attente et présente également un moyen de comprendre le fonctionnement du système en question.

Par suite, nous avons découvert les fonctionnalités de Raspberry Pi, ses types, ses nombreuses utilisations et son système d'exploitation Raspbian.

Ensuite, nous avons utilisé quelques commandes Linux de base.

En conclusion, nous avons pu programmer notre carte via Python en utilisant les bibliothèques TKinter et Socket et autres librairies.