

Rapport Mini-Projet
Architecture des Ordinateurs,

Eldis YMERAJ
Liza TOUMI

12 mars 2024



Introduction

Dans ce rapport, nous allons explorer le développement d'une application de gestion du personnel en langage assembleur x86 32 bits sous l'environnement Linux.

Dans le cadre de notre cursus, nous avons eu l'opportunité de nous plonger dans un projet pratique axé sur la programmation bas niveau. Ce projet, faisant partie de notre cours d'Architecture des Ordinateurs, visait à nous familiariser avec les concepts fondamentaux de l'assembleur et à mettre en pratique nos connaissances acquises dans un environnement réel.

Nous avons été guidés à travers une série de fonctionnalités à implémenter, destinées à couvrir divers aspects de la gestion du personnel, tels que l'enregistrement de nouveaux membres, l'affichage de la liste des personnes enregistrées, la recherche de personnes spécifiques, le calcul de l'âge moyen des personnes enregistrées, et la possibilité de quitter le programme. Ces fonctionnalités nous ont permis d'appliquer les instructions d'assembleur et de comprendre leur utilisation dans un contexte concret.

En plus de ces fonctionnalités, nous avons été confrontés à des défis de programmation spécifiques à l'assembleur, tels que la manipulation des chaînes de caractères et la gestion des boucles. Ce rapport vise à présenter de manière exhaustive notre démarche lors de ce projet, en mettant en évidence les stratégies de développement utilisées, les obstacles rencontrés et les solutions mises en œuvre.

À travers ce document, nous souhaitons démontrer notre aptitude à programmer en assembleur et à développer des solutions logicielles efficaces dans un environnement bas niveau, ce qui reflète l'importance de ces compétences dans notre parcours académique.

1 Structure du code

Dans cette section, nous allons examiner en détail la structure du code de notre application de gestion du personnel en langage assembleur x86 32 bits. Notre objectif était de créer un code bien organisé et facile à comprendre, tout en assurant une implémentation efficace des fonctionnalités requises.

1.1 Section des Données (.data)

Dans cette section, nous avons défini toutes les données statiques utilisées par notre programme. Cela comprend les messages affichés à l'utilisateur, tels que les menus, les messages d'erreur, et les messages de confirmation. Nous avons également déclaré les variables nécessaires pour stocker les informations sur les personnes enregistrées, telles que leur nom, leur âge et le nombre total de personnes enregistrées.

1.2 Section du Code (.text)

La section du code contient l'ensemble des instructions assembleur qui composent notre programme. Nous avons organisé ces instructions en fonction des différentes fonctionnalités de notre application.

- **_start** : C'est le point d'entrée principal de notre programme. À partir d'ici, nous avons appelé la fonction `display_menu` pour afficher le menu principal et ensuite géré le choix de l'utilisateur en appelant la fonction correspondante.
- **Fonctions pour chaque fonctionnalité** : Nous avons défini des fonctions séparées pour chaque fonctionnalité de notre application, telles que `enregistrer_personnel`, `lister_personnes`, `afficher_personne_specifique` etc. Chaque fonction est responsable de gérer une opération spécifique de manière isolée, ce qui rend le code modulaire et facile à maintenir.

- **Fonctions utilitaires** : En plus des fonctionnalités principales, nous avons également défini des fonctions utilitaires pour des tâches répétitives telles que l’affichage de caractères ou la réinitialisation de certains compteurs.

1.3 Commentaires Détaillés

Nous avons inclus des commentaires détaillés tout au long du code pour expliquer chaque étape et chaque instruction. Ces commentaires ont pour but d’aider les lecteurs à comprendre le fonctionnement de notre programme et la logique derrière chaque instruction.

1.4 Respect des Conventions de Nommage

Nous avons veillé à suivre des conventions de nommage claires et cohérentes pour nos variables et nos fonctions. Cela rend le code plus lisible et facilite la compréhension de sa structure par d’autres programmeurs.

En résumé, la structure du code de notre application de gestion du personnel est organisée de manière à faciliter la compréhension, la maintenance et l’extension. Nous avons adopté une approche modulaire et bien commentée, ce qui permet à chacun de naviguer facilement à travers le code et de comprendre son fonctionnement.

2 Enregistrement du Personnel

Dans cette section, nous allons examiner en détail la mise en œuvre de la fonctionnalité d’enregistrement du personnel dans notre application de gestion du personnel. Cette fonctionnalité permet à l’utilisateur d’ajouter de nouvelles personnes à la base de données, en fournissant leur nom et leur âge.

2.1 Interface Utilisateur

L’interface utilisateur pour l’enregistrement du personnel est conçue pour être simple et conviviale. Lorsque l’utilisateur choisit l’option d’enregistrement dans le menu principal, le programme affiche un message demandant à l’utilisateur d’entrer le nom et l’âge de la personne à enregistrer, séparés par un espace. Une fois que l’utilisateur a saisi les informations, il peut appuyer sur Entrée pour soumettre les données.

2.2 Traitement des Données

Une fois que l’utilisateur a soumis les données, le programme commence à traiter les informations entrées. Tout d’abord, il efface le tampon d’entrée pour éviter toute saisie précédente. Ensuite, il lit les données entrées par l’utilisateur depuis l’entrée standard.

Une fois les données lues, le programme extrait le nom et l’âge de la personne à partir de la chaîne de caractères entrée par l’utilisateur. Il utilise une boucle pour parcourir la chaîne de caractères et séparer le nom et l’âge en fonction de l’espace entre eux.

2.3 Stockage des Données

Après avoir extrait le nom et l’âge de la personne, le programme stocke ces informations dans la table des personnes. Il utilise une structure de tableau pour stocker les données de chaque personne. Chaque

enregistrement de personne comprend une entrée pour le nom et une entrée pour l'âge, avec une longueur maximale prédéfinie pour chaque champ.

2.4 Gestion des Erreurs

Le programme inclut également une gestion des erreurs pour s'assurer que les données entrées par l'utilisateur sont valides. Par exemple, si l'utilisateur entre des caractères non valides pour l'âge, le programme affiche un message d'erreur et demande à l'utilisateur de saisir à nouveau les données.

En résumé, la fonctionnalité d'enregistrement du personnel de notre application est conçue pour être simple et intuitive à utiliser. Elle permet à l'utilisateur d'ajouter de nouvelles personnes à la base de données avec facilité, tout en garantissant que les données entrées sont valides et correctement stockées.

3 Liste des Personnes Enregistrées

Dans cette section, nous décrirons en détail la fonctionnalité de liste des personnes enregistrées dans notre application de gestion du personnel. Cette fonctionnalité permet à l'utilisateur de consulter la liste complète des personnes enregistrées dans la base de données, y compris leur nom et leur âge.

3.1 Affichage de la Liste

Lorsque l'utilisateur choisit l'option de lister les personnes enregistrées dans le menu principal, le programme affiche la liste complète des personnes enregistrées sur la sortie standard. Chaque personne est affichée avec son numéro d'enregistrement, son nom et son âge, séparés par des espaces.

3.2 Traitement des Données

Pour afficher la liste des personnes enregistrées, le programme parcourt la table des personnes et extrait les informations sur chaque personne. Il commence par vérifier si la table des personnes est vide. Si c'est le cas, un message est affiché pour indiquer que la table est vide.

Si la table des personnes contient des enregistrements, le programme parcourt chaque enregistrement et extrait le numéro d'enregistrement, le nom et l'âge de chaque personne. Ces informations sont ensuite affichées à l'utilisateur.

3.3 Gestion de l'Interface Utilisateur

L'interface utilisateur pour la liste des personnes enregistrées est conçue pour être simple et conviviale. Une fois que la liste est affichée, l'utilisateur peut choisir de retourner au menu principal en appuyant sur une touche.

3.4 Exemple d'Affichage

Voici un exemple d'affichage de la liste des personnes enregistrées :

```
Liste des personnes :  
1 Eldis 22  
2 Liza 21
```

En résumé, la fonctionnalité de liste des personnes enregistrées de notre application permet à l'utilisateur de consulter facilement la liste complète des personnes enregistrées dans la base de données. Elle offre une expérience utilisateur fluide et intuitive, facilitant ainsi la gestion des données du personnel.

4 Affichage des Personnes Spécifiques

Dans cette section, nous décrirons en détail la fonctionnalité d'affichage des personnes spécifiques dans notre application de gestion du personnel. Cette fonctionnalité permet à l'utilisateur de rechercher et d'afficher les informations sur une personne enregistrée dans la base de données en saisissant son numéro d'enregistrement.

4.1 Interface Utilisateur

Lorsque l'utilisateur choisit l'option d'afficher une personne spécifique dans le menu principal, le programme demande à l'utilisateur de saisir le numéro d'enregistrement de la personne qu'il souhaite afficher. Une fois que l'utilisateur a saisi le numéro d'enregistrement, le programme affiche les informations sur la personne correspondante, y compris son nom et son âge, ou affiche un message d'erreur si aucun enregistrement correspondant n'est trouvé.

4.2 Traitement des Données

Pour afficher les informations sur une personne spécifique, le programme recherche l'enregistrement correspondant dans la table des personnes en fonction du numéro d'enregistrement saisi par l'utilisateur. S'il trouve un enregistrement correspondant, il extrait les informations sur la personne et les affiche à l'utilisateur. Sinon, il affiche un message d'erreur indiquant qu'aucun enregistrement correspondant n'a été trouvé.

4.3 Gestion de l'Interface Utilisateur

L'interface utilisateur pour l'affichage des personnes spécifiques est conçue pour être simple et conviviale. Une fois que les informations sur la personne spécifique sont affichées, l'utilisateur peut choisir de retourner au menu principal en appuyant sur une touche.

4.4 Exemple d'Affichage

Voici un exemple d'affichage des informations sur une personne spécifique :

```
Entrez le numéro identifiant de la personne a chercher: : 2
2 Eldis 22
```

Dans cet exemple, l'utilisateur a saisi le numéro d'enregistrement 2, et le programme a affiché les informations sur la personne correspondante, à savoir le nom "Bob" et l'âge "30".

En résumé, la fonctionnalité d'affichage des personnes spécifiques de notre application permet à l'utilisateur de rechercher et d'afficher facilement les informations sur une personne enregistrée dans la base de données. Elle offre une expérience utilisateur fluide et intuitive, facilitant ainsi la consultation des données du personnel.

5 Âge Moyen de Toutes les Personnes Enregistrées

Dans cette section, nous discutons de la fonctionnalité que nous avons envisagée pour calculer et afficher l'âge moyen de toutes les personnes enregistrées dans notre application de gestion du personnel. Cette fonctionnalité, essentielle pour fournir une vue d'ensemble sur la démographie du personnel enregistré, n'a malheureusement pas été intégrée dans la version actuelle de l'application en raison de divers défis rencontrés au cours du développement.

5.1 Difficultés Rencontrées

Lors de notre tentative de mise en œuvre de la fonctionnalité pour afficher l'âge moyen, nous, en tant que binôme, avons fait face à plusieurs difficultés. Celles-ci incluaient des défis liés aux erreurs de logique dans le calcul des moyennes, aux problèmes de conversion et de manipulation des données d'âge, ainsi qu'aux complications en matière d'optimisation des performances pour le traitement des données.

Nous avons adopté plusieurs approches pour essayer de résoudre ces problèmes, notamment en recherchant des solutions sur des forums de programmation, en consultant la documentation qui est disponible en ligne sur `ent`, et en testant différentes structures de données pour le stockage et la manipulation efficaces des informations d'âge. Malgré nos efforts conjoints, nous avons été incapables de surmonter ces obstacles dans les délais impartis.

6 Quitter le Programme

Dans cette section, nous aborderons la fonctionnalité de quitter le programme dans notre application de gestion du personnel. Cette fonctionnalité permet à l'utilisateur de terminer l'exécution du programme et de sortir proprement de l'application.

6.1 Interface Utilisateur

Lorsque l'utilisateur choisit l'option de quitter le programme dans le menu principal, le programme se termine immédiatement et l'exécution est interrompue. Aucune autre action n'est requise de la part de l'utilisateur.

6.2 Implémentation

L'implémentation de la fonctionnalité de quitter le programme est relativement simple. Lorsque l'utilisateur sélectionne cette option dans le menu principal, le programme utilise un appel système pour terminer proprement l'exécution. Cela garantit que toutes les ressources sont libérées et que toutes les opérations en cours sont correctement finalisées.

6.3 Impact sur l'Utilisateur

La fonctionnalité de quitter le programme offre à l'utilisateur un moyen simple et rapide de mettre fin à l'exécution de l'application. Cela lui permet de quitter l'application à tout moment, sans avoir à effectuer d'autres actions complexes.

CONCLUSION

Ce projet en langage assembleur nous a offert une occasion précieuse de plonger dans les détails les plus intimes de la programmation bas niveau, nous permettant de comprendre non seulement la complexité inhérente à la gestion directe de la mémoire et des ressources système, mais aussi la puissance et la flexibilité que cette approche peut offrir. Bien que nous ayons rencontré des défis, notamment dans l'implémentation de certaines fonctionnalités comme le calcul de l'âge moyen, l'expérience globale a été extrêmement enrichissante.

Nous avons réussi à mettre en œuvre plusieurs fonctionnalités clés de notre application de gestion du personnel, ce qui démontre notre capacité à naviguer dans l'environnement de développement assembleur et à appliquer nos connaissances théoriques dans un contexte pratique. Les obstacles rencontrés nous ont poussés à approfondir notre compréhension du langage et à développer des compétences de résolution de problèmes qui seront sans aucun doute bénéfiques pour notre future carrière d'ingénieur en informatique.

En regardant vers l'avenir, nous sommes motivés à continuer à explorer le développement en langage assembleur, à affiner nos compétences dans ce domaine et à surmonter les limitations que nous avons rencontrées dans ce projet. Nous sommes convaincus que les leçons apprises au cours de ce projet nous préparent bien à relever des défis plus complexes et à contribuer efficacement au monde de la technologie.

Difficultés rencontrées

Tout au long de ce projet, nous avons été confrontés à plusieurs défis qui ont testé notre persévérance et notre capacité à résoudre des problèmes. Le premier défi majeur a été la compréhension approfondie du fonctionnement interne du langage assembleur, notamment la gestion manuelle de la mémoire, qui est significativement différente de celle rencontrée dans les langages de programmation de haut niveau auxquels nous étions plus habitués.

De plus, la mise en œuvre de l'interface utilisateur en langage assembleur a présenté ses propres ensembles de défis, car elle nécessitait une attention particulière à la gestion des entrées et sorties, ainsi qu'à la manipulation des chaînes de caractères et des données numériques. Cela nous a obligés à réfléchir de manière créative et à rechercher des solutions innovantes pour offrir une expérience utilisateur fluide.

La tentative d'implémentation de la fonctionnalité de calcul de l'âge moyen a mis en évidence des difficultés spécifiques en matière de logique de calcul et d'optimisation des performances. Les problèmes rencontrés dans cette tâche, allant des erreurs de logique aux problèmes de performance, ont souligné l'importance de la rigueur dans la planification et le développement d'algorithmes en assembleur.

En dépit de ces défis, chaque obstacle rencontré a été une occasion d'apprentissage, nous permettant de développer une compréhension plus profonde de la programmation bas niveau et d'améliorer nos compétences en débogage et en optimisation de code. Ces expériences, bien que difficiles, ont été cruciales pour notre développement professionnel et personnel en tant que étudiants en L3 Informatique.