Rapport de

Mini Project

Sujet :

GESTION DES CANDIDATS (militaire)

Encadré par : Mr SAADI Mostafa

Réalisé par :

:SBAAI haytham (N° 190)

: GHOUNDAL youssef (N°114)

Remerciements

Ont tien tout d’abord à remercier notre encadrant Mr. SAADI Mostafa et Nous souhaitons à exprimer notre gratitude aussi à avoir donné l'occasion pour faire un mini-projet qui nous a permet d’acquérir des nouvelles connaissances et de développer nos compétences.

sommaire

INTRODUCTION GÉNÉRALE :

La gestion des données est un critère essentiel pour toute entreprise ou établissement il se peut que ça soit une gestion de stock, gestion des ressources humaines .....

pour ce projet on a opté pour le cas d'un concours de militaire qui aura besoin de gérer l'ensemble des candidats à travers une interface simple et pratique. Cette gestion sera bien sûr traitée à travers une base de données stockée sur le local ou sur un serveur privé de l'administration.

ÉTUDES DES BESOINS:

Le système comportera différentes fonctionnalités nécessaires pour une meilleure gestion.
L'application doit accomplir les traitements suivants :

- Ajouter des candidats
- Chercher et afficher des candidats
- Modifier des candidats
- Suprrimer des candidats

RÉALISATION:

Dans ce chapitre, nous présentons l'architecture sur laquelle nous avons développé notre application, les différents outils utilisés ainsi que les composantes applicatives réalisées.

Les outils nécessaires sont :

- Des structures : contiennent les informations personnelles de chaque étudiant.
- Des fonctions : pour simplifier la tâche.
- Des fichiers afin d'enregistrer les données entrées.
- Les commandes :

  - fopen: ouvrir un fichier
  - fclose :fermer un fichier
  - fprintf :pour écrire dans le fichier
  - fscanf : lire le contenu d'un fichier
  - Remove/Rename :supprimer un fichier/renommer un fichier

- fflush(stdin) :permet de libérer un tampon(BUFFER) dans la mémoire pour entrer une autre valeur.
- Strcmp : La fonction strcmp compare lexicographiquement deux tableaux de caractères à terminaison nulle. Les fonctions renvoient une valeur négative si le premier argument apparaît avant le second dans l'ordre lexicographique, zéro s'il est égal ou positif si le premier argument apparaît après le second dans l'ordre lexicographique.
- Strcpy : fait copier une chaine à une autre .

TACHES EFFECTUÉES :

Les structures :

```
typedef struct
{
    char nom[tab],prenom[tab],sexe;
    int age;
    float taille,poids;
}
Candidat;
Candidat fiche;
```

Dans le programme la structure est déclarée comme suit :
typedef struct
Nom : nom du candidat (type char).

Prénom : prénom du candidat (type char).

Age : (type entier)

Taille : (type réel).
poids:(type réel).

Candidat ;
Candidat fiche ;

fiche c'est la déclaration d'une variable de type "Candidat"

Les fonctions :

On a utilisé 12 fonctions, toutes les autres fonctions auront les mêmes arguments SAUF seek_Candidat

1. Fonction pour création de fichier : qui va contenir les informations personnelles d'un candidat et qui portera le nom,prenom,age,taille,sexe,poids de ce dernier.

On ouvre le fichier en mode écriture, puis l'utilisateur doit entrer les informations, lorsqu'il termine les données seront enregistrées dans le fichier en utilisant la commande fwrite

```
//ecrire ces information dans le fichier
fwrite(&fiche1,sizeof(Candidat),1,f);
```

Puis on ferme le fichier avec fclose.

2. Fonction pour création des informations : personnelles d'un groupe de candidats, la fonction a pour argument f de type FILE*, et 'name' le nom du fichier de type char *

```
for(i=0;i<5;i++){
    fflush(stdin);
    printf("\nle nom du candidat numero %d:",i+1);
    gets(fichegrp1[i].nom);
    fflush(stdin);
    printf("\nle prenom du candidat numero %d:",i+1);
    gets(fichegrp1[i].prenom);
    fflush(stdin);
    printf("\nla ville du candidat numero %d:",i+1);
    gets(fichegrp1[i].ville);
    // ecriture de tous les information de fichegrp1
    fwrite(&fichegrp1[i],sizeof(Groupe),1,f);
}
```

3. Fonction d'affichage : du contenu d'un fichier en ordre alphabétique, avec la commande fread, puis il affiche les informations à l'écran (à l'aide de printf est les fonctions strcmp et strcpy).

Boucle principale :

```
for(i=0;i<=nbrcandidat;i++)
    for(j=i+1;j<=nbrcandidat;j++){
        if(strcmp(str[i],str[j])>0){
            //dans cette partie si str[i] est plus grande que str[j]
            //c-à-d les chars(alphabets) de str[j] se trouves apres str[i]
            //puis une fonction d'echange simple en deux champs
            //champs pour les noms et champs pour les fiches
            strcpy(temp,str[i]);
            tempfiche1=nvlfiche2[i];
            strcpy(str[i],str[j]);
            nvlfiche2[i]=nvlfiche2[j];
            strcpy(str[j],temp);
            nvlfiche2[j]=tempfiche1;
        }
    }
```

4. Fonction d' ajout du candidat : demande à l'utilisateur d'entrée les information personnels de ce nouveau candidat on ouvre le fichier en mode "w" écrire puis on utilise les commandes printf scanf fwrite.

boucle principale :

```
printf("\nentrez le nom du candidat: ");
gets(fiche2.nom);
fflush(stdin);
printf("\nentrez le prenom du candidat: ");
gets(fiche2.prenom);
printf("\nson age:");
scanf("%d",&fiche2.age);
printf("\nla taille:");
scanf("%f",&fiche2.taille);
printf("\nle poids :");
scanf("%f",&fiche2.poids);
printf("\nle sexe(f/m):");
scanf(" %c",&fiche2.sexe);
printf("\noperation succeed.");
// ecriture de tous les information de fiche2 en
fwrite(&fiche2,sizeof(Candidat),1,f);
```

5. Fonction seek_candidat : pour rechercher à un candidat par son nom et prénom.

Boucle principale :

```
while(fread(&fiche3,sizeof(Candidat),1,f)!=0){
    //teste si le nom et prenom sont identiques
    if((strcmp(fiche3.nom,seek1name)==0)&&(strcmp(fiche3.prenom,seek2name)==0))
    {
    printf("le candidat existe:\n");
    t=fiche3.taille;
    p=fiche3.poids;
    printf("nom:%s prenom:%s age:%d taille:%f poids:%f sexe:%c",fiche3.nom,fiche3.prenom,
    // operation succeed
    return 1;
    //fermeture de fichier
    fclose(f);
    }
}
printf("le candidat n existe pas");
return 0;
```

6. Fonction est add_groupe : permet d'ajouter un groupe de candidats

Boucle d'ajout régulière.

7. Fonction de suppression : du candidat par prendre leur nom et prénom puis teste s'il existe ou non (cette opération est effectuée à l'aide d'un fichier temporaire).

boucle principale :

```
while(fread(&fiche4,sizeof(Candidat),1,f)!=0)
{
    //teste si nom et prenom sont identiques
    if((strcmp(nom1,fiche4.nom)!=0)&&(strcmp(prenom1,fiche4.prenom)!=0
    {
        //ecrire les element que nous voulont pas supprimer en temp
        fwrite(&fiche4,sizeof(Candidat),1,fichiertemp);
    }
```

8. Fonction de recherche du groupe: par son ville puis l'affiche.

boucle principale :

```
while(!feof(f)){
//lecture des information de 'f'
fread(nom1,tab,1,f);
fread(prenom1,tab,1,f);
fread(ville1,tab,1,f);
//si la ville existe et identique avec une autre ville de fichier
if((strcmp(ville2,ville1)==0))
{
// pour l'afficher l'existence une seule fois (il y a une incrementation)
if(m==0)
printf("le groupe existe: \n");
printf("Candidat %d: \n",n+1);
printf("le nom: %s\t\tle prenom: %s\t\tla ville: %s\n",nom1,prenom1,ville1);
n++;
m++;
}
```

9. Fonction de modification des informations de candidats : après vérification de est ce qu'il existe ou non (la fonction contient un menu pour changer un certaine champ dans la fiche du candidat).

Boucle principale :

```
while(fread(&fiche5,sizeof(Candidat),1,f)!=0)
{
    //teste si le nom et le prenom sont identiques
    if((strcmp(nom1,fiche5.nom)==0)&&(strcmp(prenom1,fiche5.prenom)==0)){
        //copie des informations
        strcpy(ancprenom,fiche5.prenom);
        strcpy(ancnom,fiche5.nom);
        anctaille=fiche5.taille;
        ancpoids=fiche5.poids;
        ancage=fiche5.age;
        ancsexe=fiche5.sexe;}
        // s'ils ne sont pas identiques (ne sont pas les candidats qu'on
    if((strcmp(nom1,fiche5.nom)!=0)||(strcmp(prenom1,fiche5.prenom)!=0))
        {
            //ecrire les informations de fiche5 en fichiertemp
        fwrite(&fiche5,sizeof(Candidat),1,fichiertemp);
        }
    }
```

10.Fonction main: On arrivant en programme principale, dans ce programme principale on va seulement afficher le menu à l'utilisateur afin de choisir l'opération, puis on fait l'appel aux fonctions utilisées à l'entête indépendamment les unes les autres.

```
do
{
printf("\n\t\t\tGESTION DE FICHIER\n");
printf("\t\t\t--------------------\n\n");
printf("CREATION DU FICHIER CANDIDATS --->  1\n");
printf("LECTURE DU FICHIER CANDIDATS --->  2\n");
printf("AJOUTER UNE FICHE CANDIDAT --->  3\n");
printf("RECHERCHER UNE FICHE CANDIDAT --->  4\n");
printf("CREATION DU FICHIER DES GROUPES --->  5\n");
printf("LECTURE DU FICHIER DES GROUPES --->  6\n");
printf("AJOUTER UNE FICHE GROUPE --->  7\n");
printf("SUPPRESSION D\'UN CANDIDAT --->  8\n");
printf("RECHERCHER UN GROUPE --->  9\n");
printf("modifier un candidat --->  10\n");
```

```
printf("SORTIE ---> 99\n\n");
printf("VOTRE CHOIX: ");
scanf("%d",&choix);
switch(choix)
{
case 1:create_Candidat_file(fichier,name2);break;
case 2:alphab_display(fichier,name2);break;
case 3:add_Candidat(fichier,name2);break;
case 4:
     fflush(stdin);
     printf("entrez le nom du candidat a chercher: ");
     gets(nom);
     fflush(stdin);
     printf("entrez son prenom: ");
     gets(prenom);
     seek_Candidat(fichier,name2,nom,prenom);break;
case 5:create_Groupe_file(fichier,name1);break;
case 6:view_file_Groupe2(fichier,name1);break;
case 7:add_Groupe(fichier,name1);break;
case 8:deleting_Candidat(fichier,name2);break;
case 9:seek_Groupe(fichier,name1);break;
case 10:modify_Candidat(fichier,name2);break;
}
}
while ((choix!=99));
}
```

CONCLUSION GÉNÉRALE

A travers ce projet. Nous avons découvert de nouvelles informations et réussi à produire notre programme. Ce projet d'application de gestion de candidats donne un aperçu de gestion de militaire qui permet de gérer l'ensemble des candidats et gérer leurs inscriptions , Créer de nouveaux fichier candidats et leur affecter les processus y afférant (profil(nom, prénom, age,taille,poids,sexe,...)

En effet, nous avons rencontré des difficultés et les avons surmontées, notamment : manque de temps pour mener à bien ce projet, et utilisez un ensemble des fonctions que nous n'avons pas utilisées auparavant.