MIME Type

Lilian Hiault

8 avril 2019

Table des matières

Introduction

« MIME Type »est un problème à résoudre sur Coding Games `https://www.codingame.com/training/easy/mime-type`.
On dispose d'une association de types MIME avec leurs extensions puis on reçoit des noms de fichiers dont on doit donner le type MIME.

1 Comment stocker les informations ?

1.1 Structure de types MIME

J'ai décidé de stocker les associations extensions/type MIME dans un tableau de structures dont chaque élément à un champ pour le nom du type MIME, et l'extension qui correspond.

```
// Structure qui contient le nom et l'extension d'un type MIME.
typedef struct{
  char nom[51];
  char extension[11];
}mime;
```

1.2 Tableau de types MIME

L'utilisateur entre ensuite le nombre d'éléments qui vont composer la table d'association, puis le nombre de fichiers à tester.

```
  int main()
{
    int nbAssoc; // Nombre d'éléments qui composent la table d'association.
    scanf("%d", &nbAssoc);
    int nbFichiers; // Nombre de fichiers à être analysé.
    scanf("%d", &nbFichiers);
```

On crée un tableau pour y garder les associations. Chaque case contient une structure MIME.

```
  mime * tableMime = (mime *) malloc(nbAssoc*sizeof(mime)); // Tableau qui
  // correspond à la table d'association de types MIME.
```

On récupère les associations et on les associe aux champs correspondant pour chaque type MIME entré.

```
int i;
    for (i = 0; i < nbAssoc; i++) // On récupère les associations extension/type
    {
        char extension[11]; // Extension de fichier.
        char typeMime[51]; // Type MIME.
        scanf("%s%s", extension, typeMime); fgetc(stdin);
        strcpy(tableMime[i].nom, typeMime);
        strcpy(tableMime[i].extension, extension);
    }
```

2 Retrouver le bon type

2.1 Extraire l'extension du fichier

Ensuite, l'utilisateur entre les fichiers à tester.

```
      for (i = 0; i < nbFichiers; i++)
    {
        char nomFichier[258]; // On y stocke le nom complet des fichiers entrés.
        fgets(nomFichier, 258, stdin); // One file name per line.
        strtok(nomFichier, "\n"); // On retire le retour à la ligne.
        // On  extrait l'extension.
        const char * extFichier = strrchr(nomFichier, '.'); // Extension du
        // fichier avec le point.
```

On prend une taille de 258 car le fichier peut faire jusqu'à 256 caractères auxquels on doit ajouter le « \0 »et l'entrée à la ligne « \n ».
« strtok »permet de retirer les éléments voulus, ici on enlève la nouvelle ligne.
On utilise la fonction « strrchr »qui ne garde que la partie de la chaîne de caractères qu'à partir du point.

2.2 Rechercher l'extension dans le tableau de référence

Une fois qu'on a vérifié qu'il existe bien une extension au fichier, on supprime le point. On cherche ensuite dans le tableau qui recense les types MIME donné au début du programme pour

trouver une correspondance avec l'extension puis afficher le nom qui correspond.

```
          int trouve = 0; // Booléen pour arrêter la recherche dans le tableau.
        // On vérifie qu'il existe une extension après le dernier point.
        if (extFichier != NULL)
        {
            if (extFichier + 1 != NULL)
                {
                    const char * extensionFichier = extFichier + 1; //Extension
                    // sans le point.
                    int j = 0;
                    // On recherche l'extension dans la table d'association
                    // tableMime.
                    while((trouve == 0) || (j<nbAssoc))
                    {
                      if (strcasecmp(extensionFichier, tableMime[j].extension)
                      == 0)
                      // On cherche un correspondance des extensions.
                      {
                        trouve = 1;
                        printf("%s\n", tableMime[j].nom);
                        // On renvoie le nom qui correspond à l'extension.
                      }
                      j++;
                    }
                  }
                }
```

2.3 Type inconnu

Enfin, si aucune correspondance n'a été trouvée on affiche « UNKNOWN »

```
                if (trouve == 0)
              {
                printf("UNKNOWN\n");
                // Si rien a été trouve on affiche "UNKNOWN".
              }
    }
    return 0;
}
```

Conclusion

Ce problème m'a permis d'apprendre de nouvelles fonctions de la librairie string.h et de m'améliorer sur la gestion des chaînes de caractères ? Toutefois mon programme ne fonctionne pas pour tous les tests de Coding Games seulement pour certains cas.