Algorithmique et programmation

Gestion de fichiers

R.Gosswiller

Sommaire

Scanf

Scanf

Principe

Permet de lire un input utilisateur (au clavier dans une invite de commande)
Transfère les données à une adresse mémoire : utilise des pointeurs
Renvoie 1 si tout s'est bien passé (parfois plus que 1)

Exemple

```
1  int nombre = 0;
2  int* addresseNombre = &nombre;
3  int res;
4  printf("Entrez une valeur : ");
5  res = scanf("%d", addresseNombre);
6  if (res == 1)
7    printf("Votre valeur est %d.\n", nombre);
```

Attention

Pour les chaînes de caractères, il faut faire attention à la taille allouée.

Exemple

```
1  char nom[50];
2  char prenom[50];
3  int res;
4  printf("Saisissez votre nom suivi de votre pr nom : ");
5  res = scanf("%s %s", nom, prenom);
6  if(res == 2)  //2 variables, donc on renvoie 2 si tout est ok
7      printf("Vous vous appelez %s %s, est-ce correct ?\n",
8      prenom, nom);
9  else
10     printf("Vous avez fait une erreur lors de la saisie.\n");
```

Gestion de fichiers

Principes généraux

Principes

La gestion des fichiesr se fait de manière similaire à celle de Python :

- Ouverture, lecture, écriture et fermeture de fichiers avec des fonctions dédiées
- Différents mode d'ouverture : lecture, écriture...

Principes

Se référer au cours de Python pour plus de précision sur

- Le path
- Pourquoi utiliser des fichiers et comment les nommer

Dépendances

Dépendances

Tout se fait à partir de <stdio.h> (STanDard Input/Output)

Pointeur-fichier (FILE pointer)

Principe

Les fonctions qui manipulent des fichiers le font au travers d'un pointeur.

Syntaxe

```
1   FILE *fp;
```

Le curseur

Un pointeur-fichier pointe à un endroit de la mémoire morte (disque dur) plutôt qu'à un endroit de la mémoire vive (RAM).
Ce pointeur se décale lors de l'utilisation des fonctions de manipulation des fichiers.
On appelle parfois ce type de pointeur un datastream.

Fonctions de manipulation de fichiers

Fonction d'ouverture

La fonction

fopen

Syntaxe

```
1 FILE * fopen(const char *filename, const char *mode);
2 fp=fopen("c:\\test.txt", "r");
```

Utilisation de fopen

Utilisations

Voilà les différents mode d'ouverture d'un fichier :

- r - Lecture (Read)
- w - Ecriture (Write), curseur au début, création si non-existant
- a - Ajout (Append), curseur à la fin, création si non-existant
- r+ - Lecture / Ecriture (depuis le début)
- a+ - Lecture / Ecriture (curseur à la fin)
- w+ - Lecture / Ecriture (vide le fichier)

Ce sont les mêmes qu'en Python, et que dans d'autres languages.

Fonction de fermeture

La fonction

fclose

Syntaxe

```
1 int fclose(FILE *stream);
2 fclose(fp);
```

Attention

Ouvrir un fichier déjà ouvert provoque une erreur.

Utilisation de fopen

Recommandation

```
1  fopen(...);                  //on ouvre le fichier
2  copy file -> variable;       //on range le fichier dans une variable
3  fclose;                      //on referme le fichier tout de suite
4
5  do(variable);                //on travaille sur notre variable
```

Lecture et écriture textuelle

Les fonctions

fscanf et fprintf

Syntaxe

```
1 int fscanf(FILE *stream, const char *format, ...);
2 int fprintf(FILE *stream, const char *format, ...);
```

Principe

Fonctionnent comme scanf et printf, mais en précisant le pointeur de fichier en premier.

Exemple

```
1 fp=fopen("c:\\test.txt", "w");
2 fprintf(fp, "Testing...%d\n", monEntier);
```

Attention

Utiliser ces fonctions décale le pointeur !

Fonctions additionnelles

Caractère unique

Recommandation

```
1 char fgetc (FILE *fp);
```

fgetc

Permet de lire un seul caractère à la fois

Recommandation

```
1 int fputc( char c, FILE *fp );
```

fputc

Permet d'écrire un seul caractère à la fois

Lecture et écriture binaire

Les fonctions

fread et fwrite

Syntaxe

```
1 size_t fread(void *ptr, size_t size_of_elements,
2               size_t number_of_elements, FILE *a_file);
3 size_t fwrite(const void *ptr, size_t size_of_elements,
4                size_t number_of_elements, FILE *a_file);
```

Principe

Généralement utilisé pour écrire des tableaux ou des structures

Il faut ajouter un b

S'utilise avec une ouverture de fichier spécifique : on ajoute 'b' (binary) au mode de fopen.

Exemple

```
1 FILE *fp;
2 fp=fopen("c:\\test.bin", "wb");  //wb : Write Binary
3 char x[10]="ABCDEFGHIJ";
4 fwrite(x, sizeof(x[0]), sizeof(x)/sizeof(x[0]), fp);
```

Conclusion

- On utilise fopen pour ouvrir un fichier de la bonne façon
- fscanf permet de lire, et fprintf d'écrire
- Il faut bien penser à fermer le ficheir avec fclose à la fin
- il existe certaiens méthodes pour des cas plus particuliers

Slide bonus

Le mot-clé 'static'

Le mot clé static permet de conserver une variable en dehors de son scope.

Exemple

```
1  #include<stdio.h>
2
3  int funct(){
4    static int count = 0;
5    count++;
6    return count;
7  }
8
9  int main(){
10   printf("%d ", funct());
11   printf("%d ", funct());
12   return 0;
13 }
```

A noter

Une variable static est automatiquement initialisée à 0, et est palcée dans la section mémoire plutôt que dans la stack de la RAM.
Ne pas utiliser dans une struct.

Fonction statique

En C : Une fonction statique ne peut pas être appellée en dehors de son fichier ("private" en c++)