Objectif 04
Les fichiers

Aziz Salah
salah.aziz@uqam.ca

Département d'informatique
UQÀM

Automne 2013

Opérations sur les fichiers

- ouverture
- lecture
- écriture
- positionnement
- test de fin de fichier
- fermeture

Scénario d'utilisation de fichiers

- Toute opération sur les fichiers doit être initialisée par l'opération d'ouverture et se terminer par la fermeture du fichier
- Scénario :
  1. Ouverture du fichier.
  2. Tant que la fin du fichier n'est pas atteinte, lire des données à partir du fichier.
  3. Fermeture du fichier

Ouverture de fichier selon la librairie C standard

FILE *fopen(const char *filename, const char *mode) ;

- filename : est un pointeur sur une chaine de caractères contenant le nom du fichier à ouvrir incluant possiblement le path par rapport au répertoire courant du programme en exécution.
- mode est une chaîne de caractère qui indique les opérations que l'on veut utiliser sur le fichier. Les modes les plus utilisés sont (1)mode lecture seul "r" (2)mode écriture seule "w"et (3)lecture/écriture "a+"
- FILE * est un pointeur sur un type File représentant le flux des octets du fichier ouvert. Cette valeur retournée sera utilisée par la suite pour toute les opérations sur le fichier comme lecture, écriture, fermeture etc. Si l'ouverture échoue la valeur retournée est NULL.

Les modes d'ouvertures

- "r" ouverture d'un fichier existant en lecture seule. Curseur au debut du fichier.
- "r+" ouverture d'un fichier existant en lecture/écriture. Curseur au début. Sans effacement le contenu initial.
- "w" ouverture d'un fichier en écriture seule et effacement du contenu. Fichier sera créé s'il n'existe pas.
- "w+" ouverture d'un fichier en lecture/écriture et effacement du contenu initial. Fichier doit exister
- "a" ouverture d'un fichier en écriture à la fin. Curseur à la fin. Fichier sera créé au besoin.
- "a+" ouverture d'un fichier en lecture/écriture. Curseur à la fin. Fichier sera créer au besoin.

Écriture et lecture formatée sur un fichier

int fprintf(FILE *stream, const char *format, arg1,arg2, ...) ;

- Similaire à printf
- Retourne le nombre de caractères écrits et un nombre négatif en cas d'erreur

int fscanf(FILE *stream, const char *format, arg1,arg2, ...) ;

- Similaire à scanf
- Retourne le nombre de caractères lus et un nombre négatif en cas d'erreur

Fichier ouvert par défaut

- FILE * stdin ; //entrée standard : le clavier par défaut
- FILE * stdout ; //sortie standard : l'écran pas défaut
- FILE * stderr ; //sortie des erreurs : l'écran pas défaut

- Sont équivalents
  - fprintf(stdout,format, arg1,arg2) ;
  - printf(format, arg1,arg2) ;
- Sont équivalents
  - fscanf(stdin,format, arg1,arg2) ;
  - scanf(format, arg1,arg2) ;

Ferméture d'un fichier

int fclose(FILE *stream) ;

fclose() permet de transférer les données du buffer de stream vers le fichier, le buffer du stream est détruit, stream est dissocié du fichier et le fichier est fermé.

Test de fin de fichier

int feof(FILE *stream) ;

feof() retourne :

- une valeur non nulle (vrai) si la fin du fichier (càd EOF a été lue)
- une valeur nulle (faux) si c'est pas la fin du fichier

Lire un fichier d'entiers et les afficher à l'écran

```
0  #include <stdio.h>
1  int main(){
2    FILE *a_lire;
3    int valeur;
4    a_lire= fopen("infile.txt","r");
5    while (!feof(a_lire)){
6      fscanf(a_lire,"%d",&valeur);
7      printf("%d",valeur);
8    }
9    fclose(a_lire);
10   return 0;
11 }
```

Exercice

Réécrire le contenu d'un fichier

Lire la position actuelle

long ftell(FILE *stream) ;

retourne la valeur de la position de l'indicateur de position du fichier associé à stream

Se positionner dans un fichier

int fseek(FILE *stream, long offset, int whence) ;

- déplace l'indicateur de position de offset octets
  - soit à partir du début du fichier si whence a la valeur SEEK_SET
  - soit à partir de la fin du fichier si whence a la valeur SEEK_END
  - soit à partir de la position courante si whence à la valeur SEEK_CUR
- retoune 0 en cas de succès et -1 sinon

void rewind(FILE *stream) ;

Positionne le curseur au début du stream

Autres fonctions de lecture

int fgetc(FILE *stream) ;

retourne le caractère suivant du stream et EOF si fin de fichier ou erreur

char *fgets(char *s,int n,FILE *stream) ;

- Lecture d'au plus n caractères dans s à partir de stream
- La lecture s'arrête si fin de fichier ou fin de ligne
- Retourne NULL en cas d'erreur.

Autres fonctions d'ecriture

int putc(int c, FILE *stream) ;

Écrit le caractère c sur le stream

int fputs(const char *s, FILE *stream) ;

Écrit la chaine s sur le stream

Encore un exemple

```
4   char nom[20+1];
5   int note;
6   FILE *a_lire, *a_ecrire;
7
8   a_lire=fopen("donnees.data","r");
9   a_ecrire=fopen("reverse.data","w");
10  while (!feof(a_lire)){
11   fscanf(a_lire,"%20s%d",nom,&note); //nom a une taille max de 20
12   fprintf(a_ecrire,"%d ",note);
13   fprintf(a_ecrire,"%s\n",nom);
14  }
15  fclose(a_lire);
16  fclose(a_ecrire);
```

Fichiers pour l'exemple

donnees.data

```
AA 90
BB 85
CC 75
```

reverse.data !

```
90 AA
85 BB
75 CC
75 CC
```

Conclusion

- Les fonctions les plus utilisées
  - fscanf()
  - fprintf()
  - fgetc()
  - fgets()
- Ne pas oublier de consulter les pages man pour les détails