

TP9 : Fichiers

Programmation en C (LC4)

Semaine du 26 mars 2007

1 Fichiers : les bases

1.1 La syntaxe : ouverture et fermeture

1.2 Lecture par caractère

► Exercice 1

```
void prenom(FILE * f){
    int c;
    while( ((c=fgetc(f))!='\n') && (c !=EOF))
        printf("%c",c);
}
```

► Exercice 2

```
int compte_c(FILE * f){
    int cpt=0;
    while(fgetc(f)!=EOF) cpt++;
    return cpt;
}

int compte_m(FILE * f){
    int c;
    int cpt=0;
    int on_est_sur_un_mot=0;
    while((c=fgetc(f))!=EOF)
        if(c=='\n' || c=='\n') on_est_sur_un_mot=0;
        else {
            if (!on_est_sur_un_mot) {
                on_est_sur_un_mot=1;
                cpt++;
            }
        }
    return cpt;
}

int compte_l(FILE * f){
    int c;
    int cpt=0;
    while( (c=fgetc(f))!=EOF)
        if(c=='\n')
            cpt++;
    return cpt;
}
```

► Exercice 3

```
int main (int argc , char ** argv){
    FILE* f;
    if (argc!=3 || !argv[2][0] || argv[2][1] ||
        (argv[2][0]!='c' && argv[2][0]!='m' && argv[2][0]!='l')) {
        printf("%s\n", "mauvaise_syntaxe");
        exit(1);
    }
}
```

```

if ((f = fopen(argv[1], "r"))==NULL){
    printf("erreur_lors_de_l'ouverture_de_%s\n", argv[1]);
    exit(1);
}
switch(argv[2][0]){
    case 'c':
        printf("nombre_de_lettres_:_%d\n", compte_c(f));
        break;
    case 'm':
        printf("nombre_de_mots_:_%d\n", compte_m(f));
        break;
    case 'l':
        printf("nombre_de_lignes_:_%d\n", compte_l(f));
        break;
}
if (fclose(f)==EOF){
    printf("erreur_lors_de_la_fermeture_de_%s\n", argv[1]);
    exit(1);
}
exit(0);
}

```

1.3 Écriture par caractère

► Exercice 4

```

#include <stdio.h>
#include <stdlib.h>

void copie(FILE * fs , FILE * fd){
    int c;
    while( (c=fgetc(fs)) != EOF)
        fputc(c, fd);
}

int main(int argc , char ** argv){
    FILE * source , *dst;
    if ((source = fopen("exo4.c", "r"))==NULL){
        printf("erreur_lors_de_l'ouverture_en_lecture_de_source\n");
        exit(1);
    }
    if ((dst = fopen("copie_exo4.c", "w"))==NULL){
        printf("erreur_lors_de_l'ouverture_en_ecriture_de_dst\n");
        exit(1);
    }

    copie(source , dst);

    if (fclose(source)==EOF){
        printf("erreur_lors_de_la_fermeture_de_source\n");
        exit(1);
    }
    if (fclose(dst)==EOF){
        printf("erreur_lors_de_la_fermeture_de_dst\n");
        exit(1);
    }

    exit(0);
}

```

1.4 Lecture et écriture

► Exercice 5

```

int mon_getc(FILE * f){
    char c;
    if (fread(&c,1,1,f)==0)
        return EOF;
    else
        return c;
}

int mon_putc(char c, FILE * f){
    if (fwrite(&c,1,1,f)<1)
        return EOF;
    else
        return c;
}

```

► Exercice 6

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void cree_form(int n, FILE * f, char ** mots){
    int i;
    for(i=0;i<n;i++){
        fwrite(mots[i], 1, strlen(mots[i]), f);
        fwrite("\n",1,1,f);
    }
}

int main(int argc, char ** argv){
    FILE * f;
    if (argc<2){
        printf("au_moins_un_argument\n");
        exit(1);
    }
    if ((f = fopen(argv[1], "w"))==NULL){
        printf("erreur_lors_de_l'ouverture_de_%s\n", argv[1]);
        exit(1);
    }

    if (argc>2)
        cree_form(argc-2,f, argv+2);

    if (fclose(f)==EOF){
        printf("erreur_lors_de_la_fermeture_de_%s\n", argv[1]);
        exit(1);
    }
    exit(0);
}

```

2 Entrée standard, sortie standard

► Exercice 7

```

void affiche_ecran(char *s){
    while(*s != '\0'){
        fputc(*s, stdout);
        s++;
    }
}

```

► Exercice 8

```

#include <stdio.h>
#include <stdlib.h>

void rempli(FILE * out, char ** questions) {
    int c;
    for (;*questions;*questions++) {
        printf("%s?_",*questions);
        fflush(stdout);
        // fprintf fonctionne comme printf, avec un premier argument en plus
        // pour specifier sur quel canal ecrire.
        fprintf(out,"%s:_",*questions);
        while ((c=fgetc(stdin)) != '\n')
            fputc(c,out);
        fputc('\n',out);
        fputc('\n',stdout);
    }
}

int main(int argc, char**argv) {
    char * questions[]={ "Nom", "Prenom", "age", "filier", "Adresse", NULL};
    FILE * out;
    if (argc!=2) {
        fprintf(stderr,"%s\n","Il faut passer exactement un argument.");
        exit(1);
    }

    if ((out = fopen(argv[1], "w"))==NULL){
        printf("erreur lors de l'ouverture de %s\n", argv[1]);
        exit(1);
    }
    rempli(out, questions);

    if (fclose(out)==EOF){
        printf("erreur lors de la fermeture de %s\n", argv[1]);
        exit(1);
    }
    exit(0);
}

```