

# Qu'est ce qu'un fichier ?

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Dans la machine (sur le disque dur) on a:

```
00110001 00100000 00110010 00110010
00100000 00110001 00110010 00110000
00001010 01000001 00001010 00110001
00101110 00110001
```

Comment « lire » ce fichier, c'est-à-dire par exemple mettre 1, 22 et 120 dans des variables de type **int**, A dans un variable de type **char** et 1.1 dans une variable de type **float** ?

Code ASCII:

Lettre	Code décimal	Code binaire
0	48	00110000
1	49	00110001
2	50	00110010
.	46	00101110
A	65	01000001
_ (espace)	32	00100000
linux: CR	10	00001010
Mac: LF	13	00001101
Windows: CR LF	13 et 10	0000101000001101

$$64 = 8 * 8 = 2^3 * 2^3$$
$$65 = 2^6 + 1$$

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120  
A  
1.1
```

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>  
  
Int main(){  
    int x,y,z;  
    char c;  
    float f;  
    FILE *pFi=NULL;  
  
    return 0;  
}
```

**Les fichiers sont manipulés avec des structures de type FILE**

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Flux:

```
00110001 00100000 00110010 00110010
00100000 00110001 00110010 00110000
00001010 01000001 00001010 00110001
00101110 00110001
```

**On ouvre le fichier en lecture**

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

Int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");

    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Flux

```
00110001 00100000 00110010 00110010
00100000 00110001 00110010 00110000
00001010 01000001 00001010 00110001
00101110 00110001
```

On vérifie que l'ouverture s'est bien passée  
(**TRES** important)

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

Int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n" );
        return 1;
    }

    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Flux

```
00110001 00100000 00110010 00110010
00100000 00110001 00110010 00110000
00001010 01000001 00001010 00110001
00101110 00110001
```

**Lecture de 1, 22 et 120**

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

Int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

**Le flux « avance »**

```
00110001-00100000-00110010-00110010
00100000-00110001-00110010-00110000
00001010-01000001 00001010 00110001
00101110 00110001
```

**Lecture de A**

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

Int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);

    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Flux

```
00110001-00100000-00110010-00110010
00100000-00110001-00110010-00110000
00001010-01000001-00001010-00110001
00101110-00110001
```

Lecture de 1.1

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

Int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120  
A  
1.1
```

On a fini de lire, on **ferme** le fichier

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);

    return 0;
}
```



# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

Dans le terminal, le programme affiche:

```
C4 — -bash — 56x9
[mathilde@abizarre:C4/$gcc -Wall LectureTresSimple.c ]
[mathilde@abizarre:C4/$./a.out ]
Lus: x 1 y 22 z 120 f A f 1.100000
mathilde@abizarre:C4/$
```

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);
    printf("Lus: x %d y %d z %d f %d f %f\n", x,y,z,c,f);
    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1  _ 22  _ 120
A
1.1
```

ATTENTION: toujours tester sur des fichiers  
pour vérifier que votre scanf est correct car il  
est très facile de faire des erreurs

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);
    printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

ATTENTION: toujours **tester** sur des fichiers  
pour vérifier que votre scanf est correct car il  
est très facile de faire des erreurs

```
C4 — -bash — 56x9
[mathilde@abizarre:C4/$gcc -Wall LectureTresSimple.c ]
[mathilde@abizarre:C4/$./a.out ]
Lus: x 1 y 22 z 120 c
    f 0.000000
mathilde@abizarre:C4/$
```

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);
    printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
    return 0;
}
```

## DESCRIPTION

The **scanf()** family of functions scans input according to a format, as described below. This format may contain conversion specifiers; the results from such conversions, if any, are stored through the pointer arguments. The **scanf()** function reads input from the standard input stream `stdin`, **fscanf()** reads input from the stream pointer stream, and **sscanf()** reads its input from the character string pointed to by s.

The **vfscanf()** function is analogous to `vfprintf(3)` and reads input from the stream pointer stream using a variable argument list of pointers (see `stdarg(3)`). The **vscanf()** function scans a variable argument list from the standard input and the **vsscanf()** function scans it from a string; these are analogous to the **vprintf()** and **vsprintf()** functions, respectively.

Each successive pointer argument must correspond properly with each successive conversion specifier (but see the **\*** conversion below). All conversions are introduced by the **%** (percent sign) character. The format string may also contain other characters. White space (such as blanks, tabs, or newlines) in the format string match any amount of white space, including none, in the input. Everything else matches only itself. Scanning stops when an input character does not match such a format character. Scanning also stops when an input conversion cannot be made (see below).

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

ATTENTION: toujours **tester** sur des fichiers  
pour vérifier que votre scanf est correct car il  
est très facile de faire des erreurs

```
C4 — -bash — 56x9
[mathilde@abizarre:C4/$gcc -Wall LectureTresSimple.c
[mathilde@abizarre:C4/$./a.out
Lus: x 1 y 22 z 120 c
f 0.000000
mathilde@abizarre:C4/$
```

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z, r;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    r=fscanf(pFi, "%d %d %d\n", &x, &y, &z);
    printf("nb valeurs lues par scanf %d\n", r);
    r=fscanf(pFi, "%c\n", &c);
    printf("nb valeurs lues par scanf %d\n", r);
    r=fscanf(pFi, "%f\n", &f);
    printf("nb valeurs lues par scanf %d\n", r);
    fclose(pFi);
    printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
    return 0;
}
```

# Lecture

Exemple de fichier texte (ASCII) nommé :  
fichier.txt

```
1 _ 22 _ 120
A
1.1
```

**Pour un un code robuste,  
tester ce que retourne fscanf.**

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    int x,y,z, n;
    char c;
    float f;
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    if(fscanf(pFi, "%d %d %d\n", &x, &y, &z) != 3){
        fprintf (stderr, "Erreur au scan de x, y et z\n");
        return 1;
    }

    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);
    printf("Lus: x %d y %d z %d f %d f %f\n", x,y,z,c,f);
    return 0;
}
```

# Ecriture

fichier.txt

1 \_ 22 \_ 120 \_ A \_ 1.1

En binaire

00110001 00100000 00110010 00110010  
00100000 00110001 00110010 00110000  
00100000 01000001 00100000 00110001  
00101110 00110001

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>

int main(){
    FILE *pFi=NULL;
    pFi=fopen("mon_fichier.txt", "r");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fscanf(pFi, "%d %d %d %c", &x, &y, &z);
    fscanf(pFi, "%c\n", &c);
    fscanf(pFi, "%f\n", &f);
    fclose(pFi);
    printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
    fclose(pFi);

    pFi=fopen("mon_fichier.txt", "w");
    if(pFi==NULL){
        printf ("Erreur a l'ouverture du fichier\n");
        return 1;
    }
    fprintf(pFi, "%d %d %d %c %f\n", x,y,z,c,f);
    return 0;
}
```

# Ecriture binaire

fichier.bin

??

En binaire

00000000 00000000 00000000 00000001

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>
```

```
int main(){
```

```
FILE *pFi=NULL;
```

```
pFi=fopen("mon_fichier.txt", "r");
```

```
if(pFi==NULL){
```

```
    printf ("Erreur a l'ouverture du fichier\n");
```

```
    return 1;
```

```
}
```

```
fscanf(pFi, "%d %d %d %d", &x, &y, &z);
```

```
fscanf(pFi, "%c\n", &c);
```

```
fscanf(pFi, "%f\n", &f);
```

```
fclose(pFi);
```

```
printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
```

```
fclose(pFi);
```

```
pFi=fopen("mon_fichier.bin", "w");
```

```
if(pFi==NULL){
```

```
    printf ("Erreur a l'ouverture du fichier\n");
```

```
    return 1;
```

```
}
```

```
fwrite(&x, sizeof(int), 1, pFi);
```

```
return 0;
```

```
}
```



# Ecriture binaire

fichier.bin

??

En binaire

00000000 00000000 00000000 00000001

Programme C, dans le fichier Lecture.c

```
#include <stdio.h>
```

```
int main(){
```

```
FILE *pFi=NULL;
```

```
pFi=fopen("mon_fichier.txt", "r");
```

```
if(pFi==NULL){
```

```
    printf ("Erreur a l'ouverture du fichier\n");
```

```
    return 1;
```

```
}
```

```
fscanf(pFi, "%d %d %d %d", &x, &y, &z);
```

```
fscanf(pFi, "%c\n", &c);
```

```
fscanf(pFi, "%f\n", &f);
```

```
fclose(pFi);
```

```
printf("Lus: x %d y %c z %c c %c f %f\n", x,y,z,c,f);
```

```
fclose(pFi);
```

```
pFi=fopen("mon_fichier.bin", "r");
```

```
if(pFi==NULL){
```

```
    printf ("Erreur a l'ouverture du fichier\n");
```

```
    return 1;
```

```
}
```

```
fread(&x, sizeof(int), 1, pFi);
```

```
return 0;
```

```
}
```

