

Gestion d'erreurs

Gestion d'erreurs en C

Comment vérifier si l'appel de fonction termine avec une erreur ?

Tester la valeur retournée : la description de la fonction décrit quelle valeur retournée désigne une erreur.

Gestion d'erreurs en C

Comment identifier le type d'erreur ?

Le code d'erreur : un entier **positif** défini comme une macro-constante dans le fichier

`<errno.h>`

Le même fichier en-tête contient la variable globale

`int errno`

qui prendra le numéro d'erreur quand l'appel de fonction termine avec erreur.

Gestion d'erreurs en C

- La valeur de `errno` est pertinente uniquement si effectivement la valeur de retour de la fonction indique que l'appel s'est terminé avec erreur.
- Le numéro d'erreur n'est pas remis automatiquement à 0.

```
FILE *stream = fopen( "toto.txt", "r" ) ;  
if( stream == NULL ){  
    /* errno contient le numéro d'erreur */  
    perror( "open" );  
    exit( 1 );  
}
```

la fonction `perror()` affiche le string passé en paramètre et un message d'erreur obtenu à partir de la valeur de la variable `errno`

Gestion d'erreurs en C

- La valeur de errno est pertinente uniquement si effectivement la valeur de retour de la fonction indique que l'appel s'est terminé avec erreur.

```
FILE *stream = fopen( "toto.txt", "r" ) ;  
if( errno > 0 ){ /* incorrect */  
    perror( "open" );  
    exit( 1 );  
}
```

Ce code est incorrect, errno contient le code d'erreur si la fonction fopen() échoue, par contre si fopen() retourne une valeur différente de NULL alors la valeur errno n'a aucune signification

Gestion d'erreurs en C

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

```
void perror(const char *mess)
```

affiche le message mess suivi par la description de l'erreur.

```
#include <string.h>
```

```
char *strerror(int num)
```

retourne le message associé à l'erreur num.

```
FILE *f = fopen("toto.txt", "r+");
```

```
if( f == NULL ){
```

```
    //afficher le message d'erreur
```

```
    fprintf(stderr, "%s\n", strerror( errno ) );
```

```
    exit(1);
```

```
}
```

macro-constantes prédéfinies

`__FILE__` macro-constante chaîne de caractère -- nom de fichier source

`__LINE__` macro-constante entière, le numéro de ligne de fichier source

Pour afficher un message d'erreur avec le nom du fichier source et le numéro de la ligne (approximativement) où l'erreur s'est produit :

```
FILE *f = fopen("toto.txt", "r+");
if( s == NULL ){
    fprintf(stderr, "%s in file %s in line %d\n",
               strerror( errno ),
               __FILE__, __LINE__);
    exit(1);
}
```