

file descriptor - C

MOC : [C](#),

[Pipex - 42](#)

Source : [[]]

Projets : [[]]

Tags : [#fini](#)

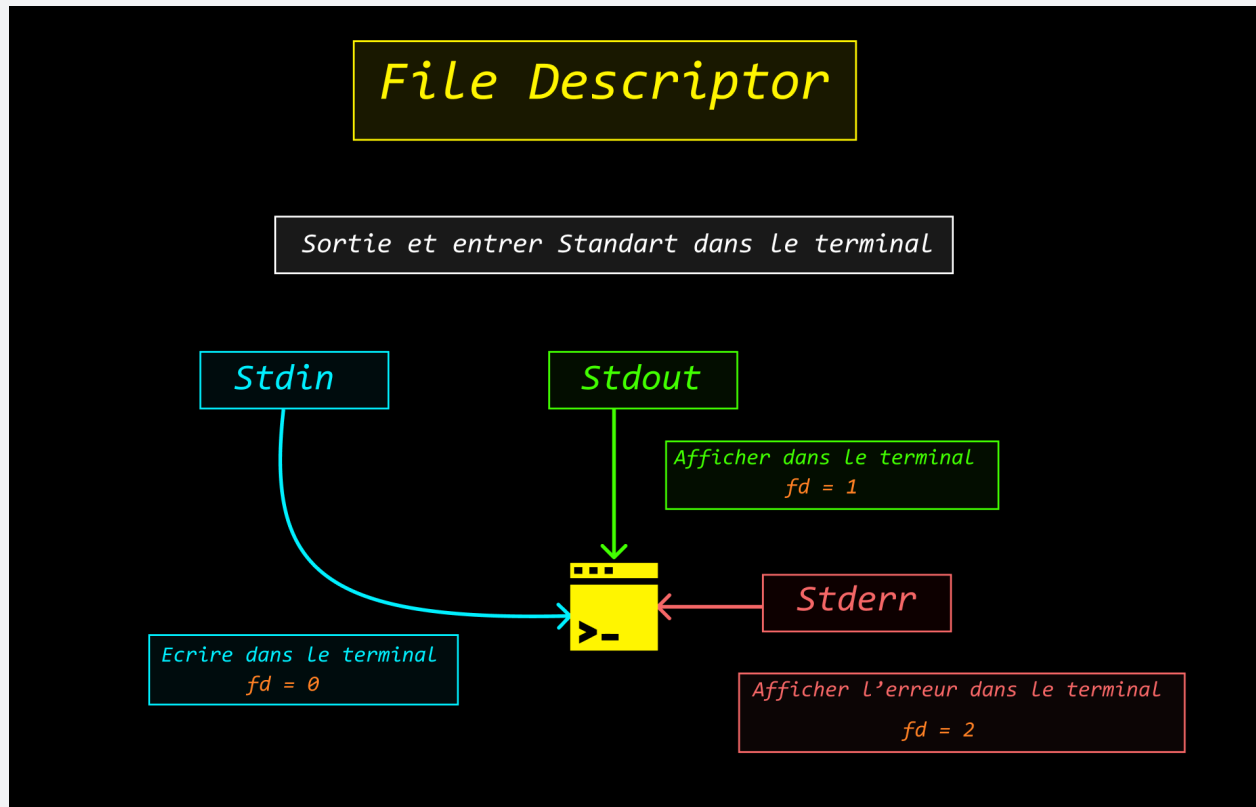
Date : 2022-05-21

Time : 14 : 23

[file descriptor - C > Creer un File Descriptor](#)

[file descriptor - C > Communiquer dans un File Descriptor](#)

Lorsqu'une commande est executer il y 3 flux Standard Assigner



[Que sont stdin, stdout et stderr sous Linux? - Azur Plus](#)

Flux de Donnees pour transferer du text

- **Le flux d'entree standart** : `STDIN_FILENO`
 - Accepte le texte comme entree
- **le flux de sortie standart** : `STDOUT_FILENO`
 - Affiche du text dans le terminal
- **le flux d'erreur standart** : `STDERR_FILENO`
 - Affiche du text dans l'erreur

Creer un File Descriptor:

- a chaques fois qu'une actions est effectuer sur un fichier comme ([Ouvrir & Lire dans un Fichier en C](#)) le fichier ce voit assigner un

`file descriptor`, un canal pour pouvoir effectuer des actions avec.

Creation de File descriptor exterieur aux terminal

Les seul *Sortie* ou *Entrer* possible reste Les *Fichiers*

1. *Ouvrir un Fichier* pour y ecrire ou y recuperer du texte



```
fd = open("test.txt")
```

```
fd = 3
```

→ 2. *Un file descriptor* superieur a 2 est creer (ex:3)

Le *fd* permet d'avoir :

- 1 *canal de sortie* pour ecrire dans un fichier
- 1 *un canal d'entrer* lire dans Le fichier

```
fd = open("test.txt", "r")
```

```
fd = 3;
```

Communiquer dans un File Descriptor :

Utiliser le fd

```
ft_putstr_fd("ton oncle\n", "File Descriptor");
```

```
ft_putstr_fd("ton oncle\n", STDIN_FILENO);
```

Stdout



```
fd = open("test.txt")  
ft_putstr_fd("ton oncle\n", fd);
```



Output

```
$ ./a.out  
>> ton oncle
```

ton oncle

Terminal :

```
ft_putstr_fd("ton oncle\n", STDOUT_FILENO);
```

```
>>> ton oncle
```

Fichier

```
int main(void)  
{  
    int fd;
```

```
fd = open("zebi.txt", O_RDONLY | O_WRONLY);  
ft_putstr_fd("ton oncle\n", fd);  
}
```

```
>>> ton oncle
```