

Système partie 1 - cours 3

Pastille 5 : fonctions de manipulation des fichiers II

N. de Rugy-Altherre - Vincent Colotte

Exemple en shell

```
cat -n fichier.txt
```

```
cat -n fichier.txt > fichier2.txt
```

Exemple en shell

```
cat -n fichier.txt
```

Le programme *cat* affiche le contenu de fichier.txt (en affichant le numéro de ligne) sur la sortie standard (écran)

```
cat -n fichier.txt > fichier2.txt
```

Exemple en shell

```
cat -n fichier.txt
```

Le programme *cat* affiche le contenu de fichier.txt (en affichant le numéro de ligne) sur la sortie standard (écran)

```
cat -n fichier.txt > fichier2.txt
```

- 1 Le système ouvre le fichier2.txt en écriture,
- 2 puis change le descripteur de sortie/écriture du processus "cat".

Exemple en shell

```
cat -n fichier.txt
```

Le programme *cat* affiche le contenu de fichier.txt (en affichant le numéro de ligne) sur la sortie standard (écran)

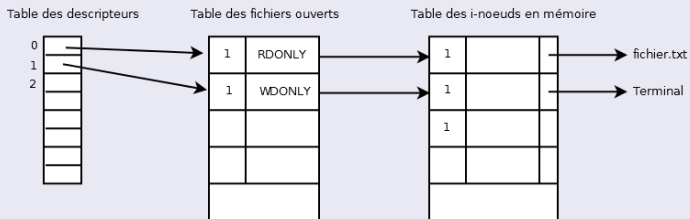
```
cat -n fichier.txt > fichier2.txt
```

- 1 Le système ouvre le fichier2.txt en écriture,
- 2 puis change le descripteur de sortie/écriture du processus "cat".

Le programme n'a pas eu besoin d'être modifié.

Exemple en shell

```
(cat -n fichier.txt)
```



```
(cat -n fichier.txt > fichier2.txt)
```

Exemple en shell

```
(cat -n fichier.txt)
```

Table des descripteurs

0	
1	
2	

Table des fichiers ouverts

1	RDONLY
1	WDONLY

Table des i-nœuds en mémoire

1			fichier.txt
1			Terminal
1			

```
(cat -n fichier.txt > fichier2.txt)
```

Table des descripteurs

0	
1	
2	

Table des fichiers ouverts

1	RDONLY
1	WDONLY
1	WDONLY

Table des i-nœuds en mémoire

1			fichier.txt
1			Terminal
1			fichier2.txt

Duplication

```
\#include <unistd.h>\\  
int dup (int $descr$) ;  
int dup2 (int $descr$, int $descr2$) ;
```

dup et dup2

Duplique le descripteur *descr*, en lui attribuant le plus petit descripteur disponible ou *descr2*.

⇒ Il crée un descripteur synonyme qui pointe vers la même entrée.

Duplication

```
\#include <unistd.h>\\  
int dup (int $descr$) ;  
int dup2 (int $descr$, int $descr2$) ;
```

dup et dup2

Duplique le descripteur *descr*, en lui attribuant le plus petit descripteur disponible ou *descr2*.

⇒ Il crée un descripteur synonyme qui pointe vers la même entrée.

Mécanisme de redirection

- Modification de l'association des descripteurs avec les fichiers physiques.
- Par exemple rediriger la sortie standard (ou d'erreur) dans un fichier sans modifier le programme initial. ⇒ La redirection est transparente pour le programme.

Bibliothèque standard C

La bibliothèque C d'entrées/sorties proposent la manipulation des fonctions précédentes à un niveau plus élevé. Le fichier n'est plus directement manipulé par un descripteur et offre ainsi une portabilité plus importante. Leur utilisation reste très proche des fonctions systèmes.

Fonctions de manipulation des fichiers :

- Appel aux fonctions système (open, read, ...) : portabilité plus importante.
 - fopen → open
 - fprintf → write
 - fscanf → read
 - fclose → close
 - ...

Bibliothèque standard C

La bibliothèque C d'entrées/sorties proposent la manipulation des fonctions précédentes à un niveau plus élevé. Le fichier n'est plus directement manipulé par un descripteur et offre ainsi une portabilité plus importante. Leur utilisation reste très proche des fonctions systèmes.

Fonctions de manipulation des fichiers :

- Appel aux fonctions système (open, read, ...) : portabilité plus importante.
 - fopen → open
 - fprintf → write
 - fscanf → read
 - fclose → close
 - ...
- Manipulation d'un objet de type FILE (lié au descripteur)
FILE file = fopen("monfichier.txt", "r");
- Permet l'utilisation de formatage
fprintf(file, "%s \n", chaine);
- Sorties standard : stdin, stdout, stderr

Bibliothèque standard C

La bibliothèque C d'entrées/sorties proposent la manipulation des fonctions précédentes à un niveau plus élevé. Le fichier n'est plus directement manipulé par un descripteur et offre ainsi une portabilité plus importante. Leur utilisation reste très proche des fonctions systèmes.

Fonctions de manipulation des fichiers :

- Appel aux fonctions système (open, read, ...) : portabilité plus importante.
 - fopen → open
 - fprintf → write
 - fscanf → read
 - fclose → close
 - ...
- Manipulation d'un objet de type FILE (lié au descripteur)
FILE file = fopen("monfichier.txt", "r");
- Permet l'utilisation de formatage
fprintf(file, "%s \n", chaine);
- Sorties standard : stdin, stdout, stderr