

Les entrées/sorties en mode console

Les noms d'unités

Les périphériques en mode console sont traités comme s'il s'agissait de fichiers. On les désigne par leur « nom d'unité » éventuellement suivi de deux points comme quand il s'agit de disques ou de disquettes (A: , B: , C: , D: ...) Un nom d'unité est un nom réservé qui s'utilise de la même manière qu'un nom de fichier.

CON: est le nom d'unité de la console. En lecture, il se rapporte au clavier ; en écriture, il fait référence à l'écran.

Pour s'en convaincre vous pouvez essayer d'afficher le contenu d'un fichier texte par la commande
COPY EXEMPLE.TXT CON

Cette commande a le même effet que la commande TYPE EXEMPLE.TXT

De même COPY CON SAISIE.TXT va enregistrer dans le fichier « SAISIE.TXT » tous les caractères que vous taperez au clavier. Il faudra terminer par le caractère [Ctrl+Z] pour mettre fin à cette saisie.

NUL: est le nom d'un fichier fictif pouvant servir de destination pour des messages que l'on veut annuler. On s'en servira lors des écritures de fichier de commandes.

Les unités **PRN, LPT1, ... AUX, COM1, ... COM4** étaient accessibles sous DOS et le sont restées en mode invite de commande jusqu'à la version 98 de Windows. L'écriture ou la lecture de ces unités est rendue impossible sur les versions ultérieures de Windows. Ces noms d'unités restent cependant réservés, il est impossible de créer un fichier qui porterait l'un de ces noms.

Les deux paragraphes suivants ne devraient donc intéresser que ceux qui utilisent le DOS ou une version de Windows antérieure à Windows 2000. Ce doit être rare !

PRN: est le nom du « fichier » imprimante. C'est le nom de la sortie parallèle initialement prévue pour y connecter une imprimante. Quand il y a plusieurs ports parallèles on les distingue par les noms **LPT1, LPT2** etc.

Si vous avez une imprimante qui fonctionne encore en mode texte connectée au port parallèle, la commande COPY EXEMPLE.TXT PRN aura le même effet que la commande PRINT EXEMPLE.TXT

AUX ou **COM1, COM2, COM3** et **COM4** sont des noms d'unités réservés aux ports séries.

Si par exemple vous avez un modem externe (avec des LED qui clignotent quand il communique) connecté sur le port COM2, faites une copie d'un fichier vers ce port pour voir la LED qui indique les bits qui passent.

COPY EXEMPLE.TXT COM2

Pour que ce test fonctionne, il faut bien sûr que le modem soit allumé, sans quoi vous aurez un message d'erreur du genre : « Erreur d'écriture sur le périphérique COM2 » ce qui prouve que ce port est aussi une entrée puisque le programme a vu que le modem est éteint.

Vous serez cependant probablement déçu de ne pas voir clignoter la LED. Avec un petit fichier texte, elle se sera juste allumée pendant une fraction de seconde. Choisissez donc un fichier assez volumineux. Faites aussi la commande MODE COM2 BAUD=150 pour ralentir la transmission et pour pouvoir percevoir les fluctuations de la LED.

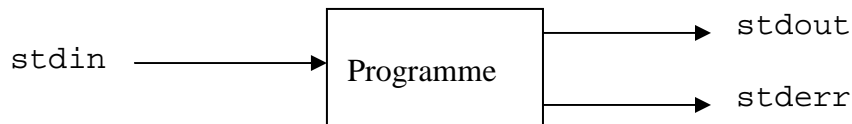
Redirection des entrées / sorties « console »

Un programme peut être vu comme un système dans lequel entrent des données, elles y sont traitées et il en sort des résultats et/ou éventuellement des signaux d'erreurs.

Les programmes lancés en mode invite de commande :

- reçoivent des données de **l'entrée standard** (généralement le clavier)
- affichent les résultats sur **la sortie standard** (généralement l'écran)
- et envoient les messages d'erreur sur **une sortie standard pour les erreurs** (généralement l'écran aussi)

En général donc, les entrées/sorties des programmes en mode invite de commande concernent la console c'est à dire le clavier et l'écran.



Mais nous avons vu que les PC possèdent d'autres « entrées/sorties ». On peut aussi imaginer que le programme sorte ses résultats dans un fichier et même qu'il puise les données à traiter dans un autre fichier encore.

Le but de ce chapitre est d'apprendre comment rediriger les entrées sorties « standards » vers d'autres entrées/sorties et vers (ou à partir) de fichiers. Cela est possible grâce au mécanisme de redirection, une particularité que Windows et UNIX ont en mode console.

Cette caractéristique peu connue se révèle parfois bien utile sans être difficile à exploiter. Elle nous permet d'envoyer la sortie console, non pas vers l'écran mais vers un fichier texte, ou même nulle part (NUL), ou de demander au programme de considérer que l'entrée provient non pas du clavier mais d'un fichier.

Redirection des sorties « stdout » et « stderr »

Pour réacheminer la sortie standard vers un fichier, il suffit de faire suivre la commande par le signe '>' suivi du nom du fichier de destination. Le symbole '>' doit vous faire penser à une flèche qui indique la sortie choisie.

Exemple : enregistrer le contenu d'un répertoire dans un fichier.

```
C:\>DIR > Directory.txt
```

Cela a pour effet de créer un fichier « Directory.txt » qui va recevoir le code sorti par la commande DIR. Ce fichier pourra ensuite être manipulé comme n'importe quel fichier texte, (Edition, recherche, impression etc.)

Si le fichier « Directory.txt » existait déjà, il serait écrasé et remplacé par un nouveau contenu.

Si vous désirez accumuler des données dans un fichier destination sans effacer ce qui y a été enregistré précédemment il suffit d'utiliser deux signes '>' consécutifs

```
Exemple : C:> Date /t >> log.txt
```

- La sortie `stdout` peut être annulée en redirigeant ce qui lui est destiné vers la sortie « NUL »

Exemple :

Pour faire une copie d'un fichier sans voir apparaître le message «1 fichier(s) copié(s)»

```
COPY Source.txt Destination.txt > NUL
```

Faites bien la différence entre la destination de la copie et la sortie standard qui devrait recevoir le message «1 fichier(s) copié(s)». La copie se fait normalement mais le message destiné à la sortie standard est annulé.

Remarquez aussi que s'il y a une erreur, le message d'erreur n'est pas annulé lui. En effet, sa destination est la « stderr » pour annuler ce type de message il faut demander la redirection de cette seconde sortie « 2> NUL »

Redirection de l'entrée standard "stdin"

Il est tout aussi facile de prendre un fichier comme origine des codes à fournir à un programme.

Exemple :

Faire un fichier qui contient des commandes destinées au programme Debug.

1° Créer un fichier Test qui contient un point d'interrogation (commande d'aide)

Et à la ligne suivante la lettre 'q' pour sortir du programme Debug

?

q

2° Lancer `C:> DEBUG < TEST`

Ces différentes redirections peuvent être combinées dans une seule commande

Exemples :

- Créer un fichier que nous appelons MEMENTO.TXT qui contient le texte d'aide des commandes du programme Debug

```
C:> DEBUG < TEST > MEMENTO.TXT
```

- Supposons que nous ayons un fichier texte appelé ANNUAIRE dont chaque ligne contient un nom suivi d'un numéro de téléphone. Pour imprimer ce fichier (sous DOS ou une ancienne version de Windows 95 ou 98 !) en le triant pour avoir les noms dans l'ordre alphabétique, il faudrait faire :

```
C:> SORT < ANNUAIRE > PRN
```

NB. 'PRN' était le nom de la sortie « imprimante »

Tubes et Filtres

Nous avons vu que les caractères '>' et '<' nous servent pour rediriger des entrées/sortie vers ou depuis des fichiers.

On pourrait imaginer un programme P1 dont le résultat sera envoyé dans un fichier TEMP qui doit servir comme entrée à un programme P2. Les commandes successives seraient :

```
P1 > TEMP
```

```
P2 < TEMP
```

Les données sorties par le programme P1 servent d'entrées au programme P2.

Il existe un moyen plus simple pour faire passer les données d'un programme à l'autre, c'est le « tube » (« pipe » en anglais) `p1 | p2`

Exemple : `C:> DIR | MORE`

Ou encore `C:> TYPE ANNUAIRE | SORT | MORE`

Les programmes qui traitent les données qui leur parviennent entrée pour les ressortir après les avoir traitées sont appelés des **filtres**.

Parmi eux, les plus célèbres sont **SORT** qui fait des tris, **MORE** qui affiche page par page et **FIND** pour faire la recherche d'une chaîne de caractères dans un fichier texte.

Exercices :

- 1 **MORE** peut servir à regrouper des lignes vides. Quelle option de cette commande faut-il utiliser ? Faites un fichier qui contient plusieurs retours à la ligne successifs puis essayez cette option de la commande **MORE**.

- 2 Editer le fichier **ANNUAIRE**

Benjamin	Franklin	Avenue de l'imprimerie 36	Tel 02 712.34.56
Yves	Coppens	Rue Lucy 10	Tel 04 311.22.33
Albert	Schweitzer	Rue du Gabon 1	Tel 04 365 76 87
Dirk	Frimout	Place de l'espace 25	Tel 010 13.24.35

- 3 Ajoutez-y les noms suivants Albert Einstein et Youri Gagarine avec des adresses et des numéros de téléphone fictifs.
- 4 Pendant que ce texte est ouvert avec le programme **EDIT**, placer le curseur sur la première lettre d'un nom de famille et noter sa position dans la ligne. (= la valeur « Col » affichée en bas de la fenêtre d'édition.)
- 5 Afficher le fichier **ANNUAIRE** à l'aide de la commande **SORT** sans utiliser d'option. Qu'est ce que ça fait ?
- 6 Utiliser la commande **SORT** pour trier les noms en faisant le tri sur les noms de famille.
- 7 Recherchez dans l'annuaire avec la commande « **FIND** », le nombre de personnes qui se prénomment **ALBERT**
Quelle commande faudrait-il utiliser pour que les noms de ces personnes s'affichent dans l'ordre alphabétique ?