Communiquer avec son programme : argc et argv

Jusqu'à présent :

- Programme exécuté sans argument
- Demande d'entrée au clavier pendant l'exécution (`scanf`)

Comment faire comme gcc, et dire quoi lui donner avant son exécution ?

```
gcc -o myprog myprog.c
```

```
$ var=a.c
$ tree
.
└── b.c
$ cp b.c a.c
```

1. Attente d'une séquence de caractère jusqu'à « Return »
2. Expansion des **variables**
3. Expansion des **chemins**
4. Est-ce une commande valide ?
5. **[Oui] Je l'éxecute**
    1. **Je passe les arguments au programme**

Shell :

« Est-ce que je connais la commande **cp** ? »

→ Oui

→ J'encapsule les arguments

→ J'invoque un **processus** de ce programme auquel je donne les arguments

Communiquer avec son programme : argc et argv

Le shell passe les arguments à la commande quoi qu'il arrive.

Si notre commande est en fait un script shell, on sait exploiter ces arguments.

```
jeremy@turing$ ./script a 6

$# : "2"
$0 : "./script.sh"
$1 : "a"
$2 : "6"
```

Communiquer avec son programme : argc et argv

Mais si c'est un programme écrit en c ?

```
#include <stdio.h>

int main() {
  return 0;
}
```

Communiquer avec son programme : argc et argv

Mais si c'est un programme écrit en c ?

```
#include <stdio.h>

int main(int argc, char *argv[]) {
   return 0;
}
```

Communiquer avec son programme : argc et argv

argv est un tableau de chaînes de caractères.

gcc -o myprog myprog.c

argc	4
argv[0]	"gcc"
argv[1]	"-o"
argv[2]	"myprog"
argv[3]	"myprog.c"

Communiquer avec son programme : argc et argv

Attention :
argc et $# sont différents

Communiquer avec son programme : argc et argv

Exercice :

```
jeremy@turing$ ./fubar a b c
This program was called with
"./fubar".
argv[1] = a
argv[2] = b
argv[3] = c
jeremy@turing$ cd dossier
jeremy@turing$ ../fubar
This program was called with
"../fubar".
The command had no other arguments.
```

Communiquer avec son programme : Fichiers

Maintenant nous savons :

- Lecture au clavier (scanf)
- Écriture à l'écran (printf)
- Arguments de la ligne de commande (entrée uniquement)
- Redirections :
  - Associer un fichier au clavier (<)
  - Associer un fichier à l'écran (>)
  - Associer la sortie d'un programme au clavier du suivant (|)

Communiquer avec son programme : Fichiers

Comment faire comme gcc, et lire dans un fichier (myprog.c) et écrire dans un autre (myprog) ?

```
gcc -o myprog myprog.c
```

Communiquer avec son programme : Fichiers

Depuis un programme C, un fichier c'est :

- Un nom « externe » (SGF, c'est une chaîne de caractères)
- Une référence interne (**descripteur** de fichier) qui notamment mémorise la position courante de lecture/écriture : **FILE ***
- Un contenu, séquence d'éléments : entiers, caractères…

Communiquer avec son programme : Fichiers

- Ouvrir le fichier : établir une association entrre le nom externe et une variable de type FILE *

```
FILE *f1, *f2 ;
f1 = fopen("fichier1", "r");
f2 = fopen("fichier2", "w");

// Tête de lecture en début
// de fichier après ouverture.
// Problème ? fopen renvoie NULL
```

Communiquer avec son programme : Fichiers

- Écrire / Lire

```
int x;
fscanf(f1, "%d", &x);
fprintf(f2, "%s", "toto");
fprintf(f2, "%d", 14+x);
```

Communiquer avec son programme : Fichiers

- Savoir quand arrêter de lire : savoir quand la tête de lecture a atteint la fin de fichier

```
feof(FILE *stream);
```

Communiquer avec son programme : Fichiers

Fermer les fichiers une fois terminé :

```
int fclose(FILE *fp);
```

Communiquer avec son programme : Fichiers

Exercice :

Un programme C qui ouvre en lecture un fichier dont le nom est passé en argument sur la ligne de commande, et affiche sa taille en octets (en comptant le nombre de caractères).

Tester les cas erreur avant.

Avant de commencer...
Petite précision sur feof.

Communiquer avec son programme : E/S standard et Redirections

```
fscanf(stdin, …);
fprintf(stdout, …);
```

stdin et stdout sont automatiquement ouverts et associés au clavier et à l'écran.

Redirection : associer stdin(out) à un fichier de l'arborescence

Le mécanisme d'écho est toujours activé, de manière indépendante.