

Passage d'arguments à un programme

Notions associées : `argc`, `argv`, `atoi` (man `atoi`)

Nous avons vu en programmation shell (Bash), qu'un shell permet de passer des arguments à un programme, lors du lancement de celui-ci. Pour mémoire, nous avons noté ces arguments `$i` ($i=0$ à n , pour n arguments passés).

`$0` désignant le nom du shell script (avec son chemin) et `$1` à `$n` les arguments passés, séparés par un espace (ou un caractère donné par la variable d'environnement `$IFS`).

De même, un programme C est capable de récupérer ces arguments qui sont stockés sous forme de chaînes de caractères. Un tableau habituellement nommé *argv*, qui doit être déclaré dans la fonction `main` contient des pointeurs sur ces chaînes. L'entier *argc* indique le nombre d'arguments passés au programme dans le tableau *argv*.

Attention : le premier argument est toujours le nom du programme lui-même (avec le chemin complet)...comme en shell

Exemple (Programme `argument.c`)

```
#include <stdio.h>

int main(int argc, char *argv[] )
{
    fprintf (stdout, "%s\n %s\n" %s\n , argv[0], argv[1], argv[2]);
    return(0);
}
```