



Université Sidi Mohamed Ben Abdellah
Faculté des Sciences Dhar Mahraz Fes
Département d'Informatiques



Compte rendue :

**Travaux dirigés de système d'exploitation N°3 :
Affichage des arguments.**

Réalisé par :

EZZOUHRI Amal

JANATI IDRISSE Meryem

Professeur : M.Meknassi



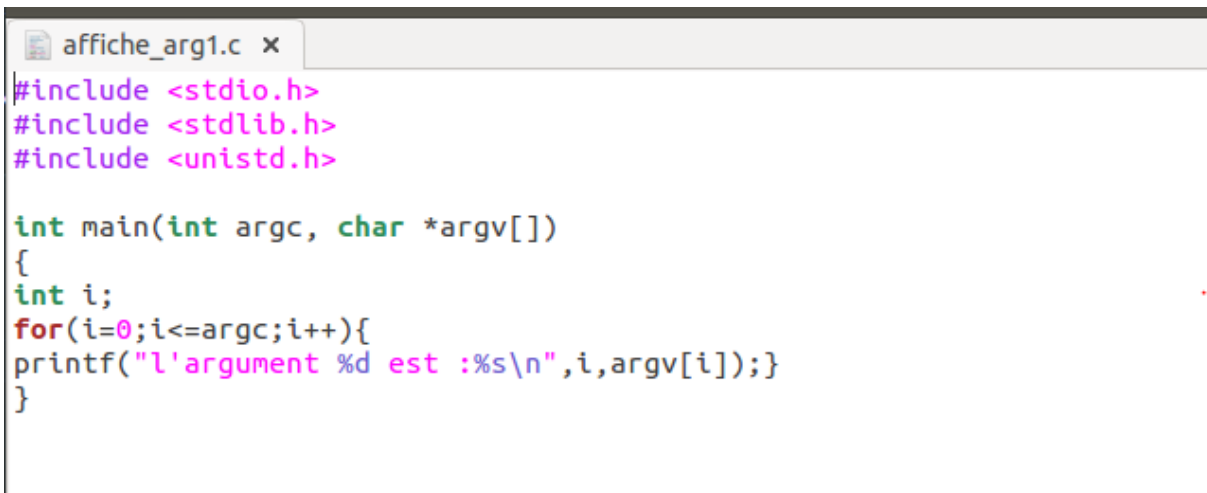
2014/2015

Introduction : Pour récupérer les arguments dans un programme en C, on utilise les paramètres `argc` et `argv` de la fonction `main`.

- `Argc` est un entier de type `int` qui donne le nombre d'arguments passés en ligne de commande plus 1.
- `Argv` est un tableau de pointeurs. `Argv[0]` contient le nom du fichier exécutable du programme. Les cases suivantes `argv[1]`, `argv[2]`, etc. contiennent les arguments passés en ligne de commande. Enfin, `argv[argc]` doit obligatoirement être `NULL`.

Enoncé : notre but dans ce rapport est d'afficher les arguments en utilisant plusieurs méthodes.

La première méthode : on fait la saisie des données en paramètre, et on les affiche à l'aide d'une boucle de 0 jusqu'à le nombre d'arguments (`argc`).



```
affiche_arg1.c x
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int i;
    for(i=0; i<=argc; i++){
        printf("l'argument %d est :%s\n", i, argv[i]);
    }
}
```

On exécute le programme ci-dessous :



```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit affiche_arg1.c
^C
ezzouhri.A@janati_idrissi.M:~$ gcc affiche_arg1.c -o affiche1.out
ezzouhri.A@janati_idrissi.M:~$ ./affiche1.out je suis en train de lire des données
l'argument 0 est :./affiche1.out
l'argument 1 est :je
l'argument 2 est :suis
l'argument 3 est :en
l'argument 4 est :train
l'argument 5 est :de
l'argument 6 est :lire
l'argument 7 est :des
l'argument 8 est :données
l'argument 9 est :(null)
ezzouhri.A@janati_idrissi.M:~$
```

La deuxième méthode : on fait la saisie des données à l'intérieur du programme.

```
*affiche_arg2.c x
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int i;
    argv[0]="./affiche2.out";
    argv[1]="je";
    argv[2]="suis";
    argv[3]="en";
    argv[4]="train";
    argv[5]="de";
    argv[6]="lire";
    argv[7]="les";
    argv[8]="données"
    argv[9]="NULL";

    for(i=0;i<=9;i++){
        printf("l'argument %d est : %s \n",i,argv[i]);
    }
}
```

Vérification :

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit affiche_arg2.c
^C
ezzouhri.A@janati_idrissi.M:~$ gcc affiche_arg2.c -o affiche2.out
ezzouhri.A@janati_idrissi.M:~$ ./affiche2.out
l'argument 0 est : ./affiche2.out
l'argument 1 est : je
l'argument 2 est : suis
l'argument 3 est : en
l'argument 4 est : train
l'argument 5 est : de
l'argument 6 est : lire
l'argument 7 est : les
l'argument 8 est : données
l'argument 9 est : NULL
ezzouhri.A@janati_idrissi.M:~$
```

La troisième méthode : on saisie les données dans un fichier shell.

```
janati@ubuntu: ~  
GNU nano 2.2.6 File: affiche_arg3.sh  
#!/bin/bash  
# decale_param.sh  
  
echo "Nombre de paramètres : $#"  
echo "Le 1er paramètre est : $1"  
echo "Le 2ème paramètre est : $2"  
echo "Le 3ème paramètre est : $3"  
echo "Le 4ème paramètre est : $4"  
echo "Le 5ème paramètre est : $5"  
echo "Le 6ème paramètre est : $6"  
echo "Le 7ème paramètre est : $7"  
echo "Le 8ème paramètre est : $8"
```

Vérification :

```
janati@ubuntu: ~  
ezzouhri.A@janati_idrissi.M:~$ pico affiche_arg3.sh  
ezzouhri.A@janati_idrissi.M:~$ chmod u+x affiche_arg3.sh  
ezzouhri.A@janati_idrissi.M:~$  
ezzouhri.A@janati_idrissi.M:~$ ./affiche_arg3.sh je suis en train de lire des données  
Nombre de paramètres : 8  
Le 1er paramètre est : je  
Le 2ème paramètre est : suis  
Le 3ème paramètre est : en  
Le 4ème paramètre est : train  
Le 5ème paramètre est : de  
Le 6ème paramètre est : lire  
Le 7ème paramètre est : des  
Le 8ème paramètre est : données  
ezzouhri.A@janati_idrissi.M:~$
```