

Rapport de Travaux Pratiques  
BTS SN-IR

---

PSYST2 – TP1 commandes unix

---

**Flavian LAXENAIRE**

17 septembre 2021

Forme / 20	[coef. 1] →		Fond / 20	[coef. 2] →		Note / 20
Qualité du rapport			Méthodologie			
Expression écrite			Respect du Cahier des Charges			
Pertinence de la rédaction			Qualité technique			
Respect des standards de codage			État d'avancement			Malus

## Sommaire

1. 1. Gestion de la ligne.....	2
2. 2. Isoler les arguments.....	2
3. 3. Lancement de la commande.....	4
4. 4. Lancement par substitution de processus.....	4

## 1. 1. Gestion de la ligne

```
int read_arg(int fd, char *line, int size)
{
    int nblus = 0; //nb caracteres lus

    //Q1 lecture ligne de commande et stockage dans line
    nblus = read(fd, line, size);
    if (nblus)
        line[nblus - 1] = 0; //mets le fin de chaine
    return nblus;
}
```

## 2. 2. Isoler les arguments

```
while (1)
{
    //Q1
    num_arg = 0;
    i = 0;

    nb = read_arg(0, line, sizeof(line)); //APPEL de read_arg()
    printf(">%s<\n", line);

    if (nb == 0)
        exit(0); //sort si CTRL-D

    list_arg[num_arg++] = line; //1er arg = commande
    char *p;
    p = line;
    int temp = 0;
    //Q2 boucle de découpage : isole les arguments de la ligne de commande
    while (i < nb)
    {
        if (line[i] == 0x20)
        {
            line[i] = 0;
            //(1 ligne ) remplace espace par 0
            //(1 ligne ) passe au caract suivant de line
            i++;
            while (line[i] == 0x20)
                i++; // boucle échappement de plusieurs espaces : ne pas retirer...

            //(1 ligne ) AFFECTE LE debut ieme argument list_arg[num_arg] = .....

            p -= temp;
            p += i;
            temp = i;
            list_arg[num_arg++] = p;
        }
        i++;

        //( 1 ligne )passe au caractere suivant de line
    }
    list_arg[num_arg] = NULL; //dernier argument
    //boucle affichage des arguments isolés
    for (i = 0; i < num_arg; i++)
    {
        printf("arg[%d] --> %s\n", i, list_arg[i]);
    }
}
```

```
snir@snir-21-22:~/Documents/bts snir 2nd annee/dev/psyst2/tp2$ ./myshell2_suj
ls -l -a
>ls -l -a<
arg[0] --> ls
arg[1] --> -l
arg[2] --> -a
```

```
snir@snir-21-22:~/Documents/bts snir 2nd annee/dev/psyst2/tp2$ ./myshell2_suj
toto titi tata tonton
>toto titi tata tonton<
arg[0] --> toto
arg[1] --> titi
arg[2] --> tata
arg[3] --> tonton
```

```
snir@snir-21-22:~/Documents/bts snir 2nd annee/dev/psyst2/tp2$ ./myshell2_suj
1 2 3 4 5 6 7 8 9 10 11 12 13
>1 2 3 4 5 6 7 8 9 10 11 12 13<
arg[0] --> 1
arg[1] --> 2
arg[2] --> 3
arg[3] --> 4
arg[4] --> 5
arg[5] --> 6
arg[6] --> 7
arg[7] --> 8
arg[8] --> 9
arg[9] --> 10
arg[10] --> 11
arg[11] --> 12
arg[12] --> 13
```

---

### 3. 3. Lancement de la commande

---

```
snir@snir-21-22:~/Documents/bts snir 2nd annee/dev/psyst2/tp2$ ./myshell2_suj
ls -la
>ls -la<
arg[0] --> ls
arg[1] --> -la
je suis le pere de pid ---> 26481
je suis le fils de pid ---> 26482
Output cmd :
total 336
drwxrwxr-x 2 snir snir 4096 sept. 10 16:46 .
drwxrwxr-x 6 snir snir 4096 sept. 17 16:06 ..
-rwxrwxr-x 1 snir snir 12760 sept. 10 16:46 myshell2_suj
-rw-rw-r-- 1 snir snir 2510 sept. 10 17:13 myshell2_suj.c
-rwxrwxr-x 1 snir snir 8384 sept. 10 16:40 test
-rw-rw-r-- 1 snir snir 311 sept. 10 16:43 test.c
-rw-rw-r-- 1 snir snir 294998 sept. 10 12:23 TP2myshell.pdf
```

---

### 4. 4. Lancement par substitution de processus

---

```
int pid = fork();
if (pid == 0) //fils
{
    //execl("/bin/ps", "ps", "-aux", 0);
    printf("je suis le fils de pid ---> %d\n", getpid());
    sprintf(cmd, "/bin/%s", list_arg[0]);
    printf("Output cmd :\n");
    execv(cmd, list_arg);
}
else
{
    printf("je suis le pere de pid ---> %d\n", getpid());
    wait(NULL);
}
```