

Programmation Impérative II

A quoi sert la programmation?

- indispensable pour plusieurs métiers.
- passion.

But du cours:

- suite du cours Prog. Imp. I
- fichiers, pointeurs, modularité et compilation séparée, allocation de mémoire.

Outils utilisés:

- openoffice, gedit, emacs, gcc (linux)
- recommandation: machine virtuelle de linux ([virtualbox](#))

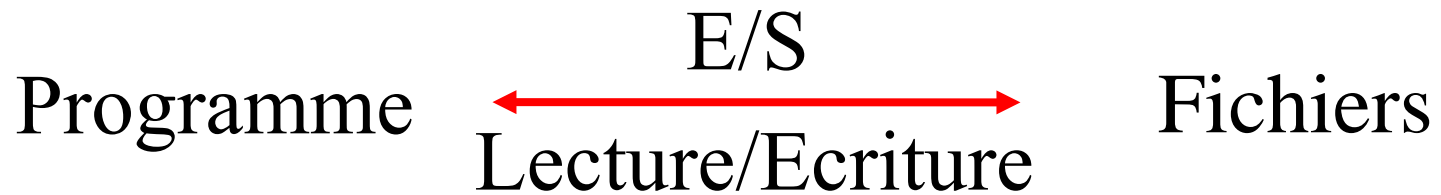
minh-anh.tran@u-pec.fr

Programmation Impérative II

Entrées / Sorties sur les fichiers

minh-anh.tran@u-pec.fr

On communique avec un programme par l'intermédiaire des fichiers



Périphériques (clavier, écran, ...) sont des fichiers

`#include <stdio.h>`

une bibliothèque
prédéfinie

Lecture à partir du fichier `stdin`

```
scanf ( "%d " , &x);  
c = getchar ();
```

Écriture sur le fichier `stdout`

```
printf ( "%d " , x);  
putchar ('a');
```

les données sont
lues à partir du
fichier `donnee`

Rédirection :

```
./executable < donnee > resultats
```

écrites dans le
fichier `resultats`

FICHER dans un programme

- Déclaration
- Ouverture
- Traitement
- Fermeture

FICHER dans un programme

Déclaration: type FILE

```
FILE *NomF ;
```

NomF est un pointeur
de type FILE
désignation d'un fichier
dans le programme

FICHIER dans un programme

Ouverture : **fopen**

```
NomF = fopen ( "/data/toto.txt" , "r" ) ;
```



Chemin d'accès



Mode: lecture

Ouverture en lecture du fichier dont le chemin d'accès est donné en premier paramètre

FICHIER dans un programme

Ouverture : **fopen**

```
NomF = fopen ( "./data/toto.txt" , "w" );
```




Mode: écriture

si fichier existe alors le contenu est perdu
sinon il est créé automatiquement

FICHIER dans un programme

Ouverture : **fopen**

```
NomF = fopen ( "./data/toto.txt" , "a" );
```



Ajout à la fin

si **NomF == NULL** alors erreur (fichier non existant, droit d'accès insuffisant, création pas possible...)
sinon on peut accéder au premier traitement

Traitements

Écriture : **fprintf, fputc**

```
fprintf (NomF, "%d", x);
```

```
fputc ('a', NomF);
```

Lecture : **fscanf, fgetc**

```
fscanf (NomF, "%d", &x);
```

```
c = fgetc (NomF);
```

```
fprintf (stdout, "%d", x); ↔ printf ("%d", x);
```

```
fscanf (stdin, "%d", &x); ↔ scanf ("%d", &x);
```

```
c = fgetc (stdin); ↔ c = getchar ();
```

```
fputc ('a', stdout); ↔ putchar ('a');
```

Détection de la fin du fichier : **feof**

vrai si c'est la fin du fichier

```
while (!feof(NomF))  
{  
    c = fgetc (NomF);  
}
```

Fermeture : **fclose**

```
fclose (NomF);
```

Exemple

```
int main (void)
{
    char CHEMIN[20], c;
    FILE * NomFE, * NomFS;

    printf ("entrer le nom du fichier a lire\n");
    scanf ("%s", CHEMIN);

    NomFE = fopen (CHEMIN, "r");

    if (NomFE == NULL)
    {
        printf ("ouverture r pas possible\n");
        exit (1);
    }
}
```

```
printf ("entrer le nom du fichier a ecrire\n");
scanf ("%s", CHEMIN);
NomFS = fopen (CHEMIN, "w");

if (NomFS == NULL)
{
    printf ("ouverture w pas possible\n");
    exit (2);
}

while (!feof(NomFE))
{
    c = fgetc (NomFE);
    fprintf (NomFS, "%c-",c);
}
fclose (NomFS);
fclose (NomFE);
return 0;
}
```