

Les chaînes de caractères

Librairie <string.h>

- Écrite en C
- Fournie des fonctions pour manipuler des chaînes de caractères
- Ne devrait pas être utilisée en C++ (sauf bonnes raisons)
- Survole rapide

Retour sur les pointeurs(?)

Traitement de chaînes

- `size_t strlen(const char* str) <- Sans '\0'`
 - opérateurs `sizeof(variable/type) <- Avec '\0'`
- `char* strcpy(char* destination, const char* source)`
- `char* strncpy(char* destination, const char* source, size_t maxlen)`
- `char* strcat(char* destination, const char* source)`
- `int strcmp(const char* s1, const char* s2)`
- `int strncmp(char* s1, const char* s2, size_t maxlen)`
- `char* strchr(const char* s, char c)`
- ...

Tests de caractères

- `int isalnum(int c)`
- `int isalpha(int c)`
- `int isgraph(int c)`
- `int isspace(int c)`

Conversions

- `int toascii(int c)`
- `int tolower(int c)`
- `double atof(const char* s)`
- `int atoi(const char* s)`

Librairie <string>

- Fait partie de la librairie standard
- S'occupe de l'allocation de la mémoire comme une grande
- Plus facile à utiliser et manipuler
- Plus claire

** Dans les exemples suivants, 'mystring' représentera un instance de la classe `std::string` selon :

```
std::string mystring("Hello");
```

Traitement de chaînes

- Longueur : `mystring.size()`
- Copie : `std::string copy = mystring;`
- Copie « N » : `std::string copy = mystring.substr(n)`
- Concaténation : `std::string final = s1 + s2`
- Comparaison : `s1 == s2`
- Recherche : `mystring.find()`, `mystring.rfind()`

Redirection

Les flux

- Entrée :
 - C : stdin
 - C++ : cin
 - Descripteur : 0
- Sortie standard
 - C : stdout
 - C++ : cout
 - Descripteur : 1

Les flux(2)

- Sortie des erreurs
 - C : stderr
 - C++ : cerr
 - Descripteur : 0

Principe

- Capturer la sortie d'un fichier pour la rediriger en entrée vers un autre fichier

Shell

- Ouvrir MSYS2 sous Windows
- Ouvrir le terminal sous Linux/Macintosh
- Taper
 - `echo $SHELL`
- Manifestez-vous si « tsch » est affiché au lieu de « bash »

Direction standard

- `test.exe > sortie.txt`
- Création du fichier `sortie.txt`
- Écrasement si le fichier existe déjà
- `sortie.txt` contient maintenant les messages de la sortie standard du programme `test.exe`
- `test.exe >> sortie.txt`
 - Pour éviter l'écrasement
 - Concaténera le nouveau contenu à l'ancien

Création du fichier de test

Redirection des autres flux

- Utilisation du descripteur du flux avec la commande de redirection
- `test.exe 2> err.txt`

- Redirection de la sortie standard et la sortie des erreurs
- `(test.exe > sortie.txt) 2> err.txt`

Redirection de tous les flux sortants

- Pour rediriger la sortie des erreurs et la sortie standard il suffit d'appeler la redirection avec les deux descripteurs
- `text.exe > latotale.txt 2>&1`