ALGORITHMIQUE ET PROGRAMMATION C

CHAP 4: LES CHAÎNES DE CARACTÈRES

Dr. Ikbal Chammakhi Msadaa

Représentation des chaînes de caractères

- En langage C, il n'existe pas de véritable type chaîne dans la mesure où l'on ne peut pas déclarer des variables d'un tel type.
- Il existe toutefois **une convention** de représentation des chaînes. Celle-ci est utilisée:
  - par le compilateur pour représenter les chaînes constantes (notées entre `` ``) ;
  - par un certain nombre de fonctions qui permettent de réaliser :
    - les lectures ou écritures de chaînes ;
    - les traitements classiques tels que concaténation, recopie, comparaison, extraction de sous-chaîne, conversions...
- Comme il n'existe pas de variables de types chaînes, un tableau de caractères est utilisé.

Représentation des chaînes de caractères

- En C, une chaîne de caractères est représentée par une suite d'octets correspondant à chacun de ses caractères, le tout terminé par un **caractère nul (\0)**. Une chaîne de **n caractères** occupe donc un emplacement mémoire de **n+1 octets**.
- Syntaxe: char nom_chaine[taille_chaine] ;
- "bonjour" est équivalent au tableau

b	o	n	j	o	u	r	\0

Initialisation des chaînes

```
char ch[20] ;
```

~~`ch = "bonjour" ;`~~

```
char ch[20] = "bonjour" ;
```

```
char ch[20] = { 'b','o','n','j','o','u','r','\0' }
```

- Puisque C autorise l'omission de la dimension d'un tableau: `char message[] = "bonjour" ;` cette instruction réserve un tableau de 8 caractères (compte tenu du 0 de fin).

Lecture et écriture des chaînes

- Le langage C offre plusieurs possibilités de lecture ou d'écriture de chaînes :
  - l'utilisation du code de format %s dans les fonctions printf et scanf ;
  - les fonctions spécifiques de lecture (gets) ou d'affichage (puts) d'une chaîne (une seule à la fois).

Lecture et écriture des chaînes

```
#include <stdio.h>
main()
{  char nom[20], prenom[20], ville[25] ;
   printf ("quelle est votre ville : ") ;
   gets (ville) ;

   printf ("donnez votre nom et votre prénom : ") ;
   scanf ("%s %s", nom, prenom) ;
   printf ("bonjour cher %s %s qui habitez à ", prenom, nom) ;
   puts (ville) ;
}
```

- printf et scanf permettent d'afficher plusieurs chaînes à la fois. puts et gets ne traitent qu'une chaîne à la fois.

Lecture et écriture des chaînes

- **Remarque:**

  Dans les appels des fonctions scanf et puts, les identificateurs de tableau comme nom, prenom ou ville n'ont pas besoin d'être précédés de l'opérateur & **puisqu'ils représentent déjà des adresses**. La norme prévoit toutefois que si l'on applique l'opérateur & à un nom de tableau, on obtient l'adresse du tableau. Autrement dit, **&nom est équivalent à nom**.

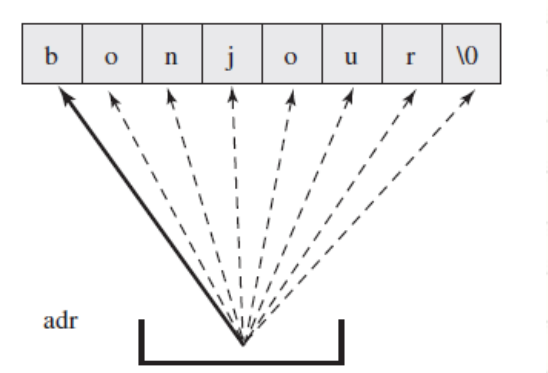
Pointeurs et chaînes de caractères

```
#include <stdio.h>
main()
{   char * adr ;                                          bonjour
    adr = "bonjour" ;
    while (*adr)
     { printf ("%c", *adr) ;
       adr++ ;
     }
}
```

- La déclaration: char * adr; réserve l'emplacement pour un pointeur sur un caractère (ou une suite de caractères).
- Pour l'affectation adr = "bonjour"; bonjour a comme valeur non pas la valeur de la chaîne elle-même mais son adresse.

Pointeurs et chaînes de caractères

- Si l'on considère le schéma illustrant ce cas de figure:
  - le trait plein correspond à la situation après exécution de l'affectation adr = "bonjour";
  - les autres flèches correspondent à l'évolution de la valeur de adr, au cours de la boucle.



Pointeurs et chaînes de caractères

- Initialisation de tableaux de pointeurs sur des chaînes de caractères.
  - Une chaîne constante est traduite par le compilateur en une adresse que l'on peut affecter à un pointeur sur une chaîne.
  - Ceci peut être généralisé à un tableau de pointeurs comme suit:

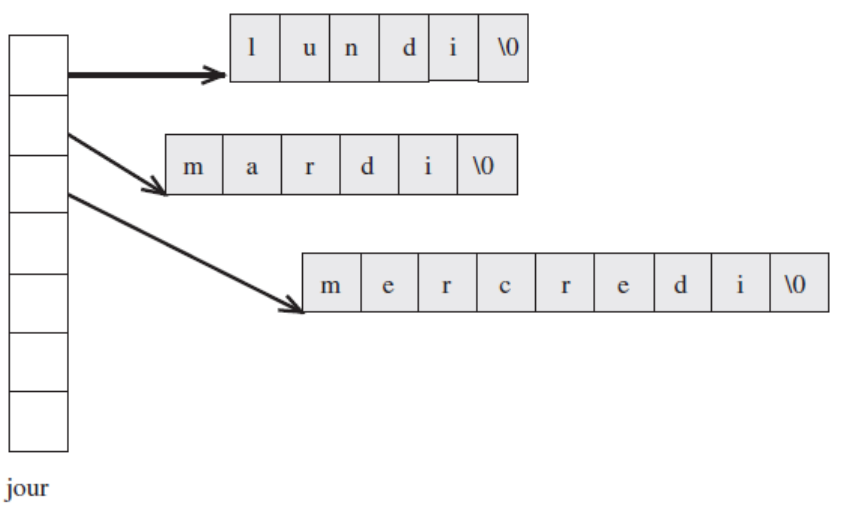
```
char * jour[7] = { "lundi", "mardi", "mercredi", "jeudi",
                  "vendredi", "samedi", "dimanche" } ;
```

Cette déclaration crée 7 chaînes constantes et initialise le tableau jour avec les 7 adresses de ces chaînes;

Pointeurs et chaînes de caractères

```
main()
{   char * jour[7] = { "lundi", "mardi", "mercredi", "jeudi",
                       "vendredi", "samedi", "dimanche" } ;
    int i ;
    printf ("donnez un entier entre 1 et 7 : ") ;
    scanf ("%d", &i) ;
    printf ("le jour numéro %d de la semaine est %s", i, jour[i-1] ) ;
}
```

```
donnez un entier entre 1 et 7 : 3
le jour numéro 3 de la semaine est mercredi
```



Les fonctions portant sur les chaînes

- La fonction **strlen**
  - **strlen** fournit en résultat la longueur d'une chaîne de caractères fournie en argument. Le caractère de code nul n'est pas pris en compte dans la longueur.
    - strlen ("bonjour") vaudra 7 ; de même, avec :
    - char * adr = "salut" ;
    - l'expression : strlen(adr) vaudra 5.

Les fonctions portant sur les chaînes

- **Les fonctions de concaténation**
  - Il existe des fonctions dites de concaténation, c'est-à-dire de mise bout à bout de deux chaînes. A priori, de telles fonctions créent une nouvelle chaîne à partir de deux autres. Elles devraient donc recevoir en argument trois adresses !
  - C a prévu de se limiter à deux adresses en convenant arbitrairement que la chaîne résultante serait obtenue en ajoutant la seconde à la fin de la première, laquelle se trouve donc détruite en tant que chaîne (en fait, seul son \0 de fin a disparu...). Là encore, on trouvera deux variantes dont l'une permet de limiter la longueur de la chaîne résultante.

Les fonctions portant sur les chaînes

Les fonctions de concaténation

- La fonction **strcat**
  - L'appel de strcat se présente ainsi **(string.h)**:

strcat (but, source)

  - Cette fonction recopie la seconde chaîne (source) à la suite de la première (but), après en avoir effacé le caractère de fin.

```
#include <stdio.h>
#include <string.h>
main()
{
   char ch1[50] = "bonjour" ;
   char * ch2 = " monsieur" ;
   printf ("avant : %s\n", ch1) ;
   strcat (ch1, ch2) ;
   printf ("après : %s", ch1) ;
}
```

```
avant : bonjour
après : bonjour monsieur
```

Les fonctions portant sur les chaînes

Les fonctions de concaténation

- **strcat** fournit en résultat :
  - l'adresse de la chaîne correspondant à la concaténation des deux chaînes fournies en argument,
    - lorsque l'opération s'est bien déroulée ; cette adresse n'est rien d'autre que celle de ch1 (laquelle n'a pas été modifiée – c'est d'ailleurs une constante pointeur),
    - le pointeur nul lorsque l'opération s'est mal déroulée.
  - Il est nécessaire que l'emplacement réservé pour la première chaîne soit suffisant pour y recevoir la partie à lui concaténer.

Les fonctions portant sur les chaînes

Les fonctions de concaténation

- La fonction **strncat**
  - Cette fonction dont l'appel se présente ainsi **(string.h)**:

strncat (but, source, lgmax)

  - travaille de façon semblable à strcat en offrant en outre un contrôle sur le nombre de caractères qui seront concaténés à la chaîne d'arrivée (but).

```
#include <stdio.h>
#include <string.h>
main()
{
   char ch1[50] = "bonjour" ;
   char * ch2 = " monsieur" ;
   printf ("avant : %s\n", ch1) ;
   strncat (ch1, ch2, 6) ;
   printf ("après : %s", ch1) ;
}
```

```
avant : bonjour
après : bonjour monsi
```

Les fonctions portant sur les chaînes

Les fonctions de comparaison de chaînes

- La fonction **strcmp**:

  strcmp (chaîne1, chaîne2)

  – compare deux chaînes dont on lui fournit l'adresse et elle fournit une valeur entière définie comme étant :
    - positive si chaîne1 > chaîne2 (c'est-à-dire si chaîne1 arrive après chaîne2, au sens de l'ordre défini par le code des caractères) ;
    - nulle si chaîne1 = chaîne2 (c'est-à-dire si ces deux chaînes contiennent exactement la même suite de caractères) ;
    - négative si chaîne1 < chaîne2

  – Exemple:
    - strcmp ("bonjour", "monsieur") est négatif
    - strcmp ("paris2", "paris10") est positif.

Les fonctions portant sur les chaînes

Les fonctions de comparaison de chaînes

- La fonction **strncmp**:

  strncmp (chaîne1, chaîne2, lgmax)

  – travaille comme strcmp mais elle limite la comparaison au nombre maximal de caractères indiqués par l'entier lgmax.

  – Exemple:

  - strncmp ("bonjour", "bon", 4) est positif
  - strncmp ("bonjour", "bon", 2) vaut zéro.

Les fonctions portant sur les chaînes

Les fonctions de comparaison de chaînes

- Les fonctions **stricmp** et **strnicmp** (***string.h***)

  stricmp (chaîne1, chaîne2)

  strnicmp (chaîne1, chaîne2, lgmax)

  – travaillent respectivement comme strcmp et strncmp, mais sans tenir compte de la différence entre majuscules et minuscules (pour les seuls caractères alphabétiques).

Les fonctions portant sur les chaînes

Les fonctions de copie de chaînes

- La fonction **strcpy**

strcpy (but, source)

  - **strcpy** recopie la chaîne située à l'adresse source dans l'emplacement d'adresse destination.
  - Là encore, il est nécessaire que la taille du second emplacement soit suffisante pour accueillir la chaîne à recopier, sous peine d'écrasement intempestif.
  - Cette fonction fournit comme résultat l'adresse de la chaîne but.

Les fonctions portant sur les chaînes

Les fonctions de copie de chaînes

- La fonction **strncpy**

strncpy (but, source, lgmax)

  - **strncpy** procède de manière analogue à strcpy, en limitant la recopie au nombre de caractères précisés par l'expression entière lgmax.
  - Notez bien que, si la longueur de la chaîne source est inférieure à cette longueur maximale, son caractère de fin (\0) sera effectivement recopié. Mais, dans le cas contraire, il ne le sera pas.

Les fonctions portant sur les chaînes

Les fonctions de copie de chaînes

```
#include <stdio.h>
#include <string.h>
main()
{
   char ch1[20] = "xxxxxxxxxxxxxxxxxxx" ;
   char ch2[20] ;
   printf ("donnez un mot : ") ;
   gets (ch2) ;
   strncpy (ch1, ch2, 6) ;
   printf ("%s", ch1) ;
}
```

```
donnez un mot : bon
bon
_______
donnez un mot : bonjour
bonjouxxxxxxxxxxxxx
```

Les fonctions portant sur les chaînes

Les fonctions de recherche dans une chaîne

- En langage C, il existe des fonctions classiques de recherche de l'occurrence dans une chaîne d'un caractère ou d'une autre chaîne (nommée alors sous-chaîne).
- Elles fournissent comme résultat un pointeur de type char * sur l'information cherchée en cas de succès, et le pointeur nul dans le cas contraire.

Les fonctions portant sur les chaînes

Les fonctions de recherche dans une chaîne

strchr (chaîne, caractère)

- recherche, dans chaîne, la première position où apparaît le caractère mentionné.

strrchr (chaîne, caractère)

- réalise le même traitement que strchr, mais en explorant la chaîne concernée à partir de la fin. Elle fournit donc la dernière occurrence du caractère mentionné.

strstr (chaîne, sous-chaîne)

- recherche, dans chaîne, la première occurrence complète de la sous-chaîne mentionnée.