



Bases de la programmation

Séance 11

Chaines de caractères et structures

Sébastien Combéfis

mercredi 10 décembre 2014



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Pas de Modification 4.0 International.

Rappels du cours précédent

- Fichier d'entête
 - Définition de constantes
 - Définition de types
 - Définitions de prototypes
- Chaîne de compilation
- Définition d'une nouvelle librairie
 - Fichier .h
 - Fichier .c

Chaine de caractères

- Une chaîne de caractères est un tableau de caractères
- Doit se terminer par le caractère terminateur `'\0'`

```
int main()  
{  
    char str[5];  
    str[0] = 'J';  
    str[1] = 'O';  
    str[2] = 'H';  
    str[3] = 'N';  
    str[4] = '\0';  
  
    printf ("%s\n", str);  
  
    return 0;  
}
```

Notations raccourcies

- Par **initialisation du tableau**

```
char str[] = { 'J', 'O', 'H', 'N', '\0' };  
printf ("%s\n", str);
```

- Avec l'utilisation de **guillemets doubles** ("")

```
char *str = "JOHN";  
printf ("%s\n", str);
```

Longueur d'une chaîne de caractères

- Parcours de la chaîne jusqu'à trouver `'\0'`

```
int getSize (char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
        i++;
    }
    return i;
}
```

```
int getSize (char *str)
{
    int size = 0;
    while (*str != '\0')
    {
        size++;
        str++;
    }
    return size;
}
```

Copie d'une chaîne de caractères

- Copie caractère par caractère jusqu'à atteindre `'\0'`

```
char* copyString (char *str)
{
    int size = getSize (str) + 1;
    char *copy = (char*) malloc (size * sizeof (char));

    int i;
    for (i = 0; i < size; i++)
    {
        copy[i] = str[i];
    }

    return copy;
}
```

Opérations sur les chaînes de caractères I

- Utilisation de la **bibliothèque** `string.h`
- Obtenir la **longueur**

```
unsigned long size = strlen (str);  
printf ("%lu\n", size);
```

- Faire une **copie**

```
char *dest = (char*) malloc (size * sizeof (char));  
strcpy (dest, str);  
printf ("%s\n", dest);
```


Opérations sur les chaînes de caractères II

■ Comparer deux chaînes de caractères

```
char *s1 = "Hello";  
char *s2 = "World";  
  
int result = strcmp (s1, s2);  
printf ("%d\n", result);
```

■ Rechercher une sous-chaîne

```
char *s1 = "Hello";  
char *s2 = "ll";  
  
char *pos = strstr (s1, s2);  
printf ("%s\n", pos);
```

Définition d'une structure

- **Type de données** composé à partir d'autres types
- Définition d'une **structure** avec le mot réservé **struct**
 - **Nom** de la structure
 - **Champs** de la structure (déclaration de variables)

```
struct date {  
    int day;  
    int month;  
    int year;  
};
```

Initialisation d'une structure

- Accès aux champs de la structure avec l'opérateur d'accès (.)
- Initialisation d'une structure champ par champ

```
struct date now;  
  
now.day = 10;  
now.month = 12;  
now.year = 2014;
```

1016	2014	(now.year)
1008	12	(now.month)
p : 1000	10	(now.day)

Copie d'une structure

- Une structure est une variable contenant **plusieurs données**
- **Copie** de tous les champs de la structure

```
struct date now;  
now.day = 10;  
now.month = 12;  
now.year = 2014;  
  
struct date d = now;      // Copie de la variable now  
d.day++;  
d.month = 2;  
d.year -= 14;  
  
printf ("%d/%d/%d\n", now.day, now.month, now.year);  
printf ("%d/%d/%d\n", d.day, d.month, d.year);
```

Nouveau type à partir d'une structure

- Nom raccourci d'une structure via **typedef**

```
typedef struct date {  
    int day;  
    int month;  
    int year;  
} DATE;
```

- Utilisation complètement similaire

```
DATE newyear;  
  
newyear.day = 1;  
newyear.month = 1;  
newyear.year = 2015;
```

Structure en mémoire dynamique

- Données d'une structure sur la **pile ou dans le tas**
- Création en mémoire dynamique avec malloc
- Accès aux champs avec **l'opérateur ->**

```
struct date *now =  
    (struct date*) malloc (sizeof (struct date));  
  
now->day = 10;  
now->month = 12;  
now->year = 2014;
```

Équivalence de notations

- Soit la déclaration de variable **struct** xxx *data,
les notations suivantes sont équivalentes :

data->champ (*data).champ

Paramètre de type structure I

```
#include <stdio.h>

void printDate (struct date d)
{
    printf ("%d/%d/%d\n", d.day, d.month, d.year);
}

int main()
{
    struct date now;
    now.day = 10;
    now.month = 12;
    now.year = 2014;

    printdate (now);

    return 0;
}
```


Paramètre de type structure II

```
#include <stdio.h>

void printDate (struct date *d)
{
    printf ("%d/%d/%d\n", d->day, d->month, d->year);
}

int main()
{
    struct date now;
    now.day = 10;
    now.month = 12;
    now.year = 2014;

    printdate (&now);

    return 0;
}
```

Renvoi d'une nouvelle structure

```
DATE* createDate (int d, int m, int y)
{
    DATE *date = (DATE*) malloc (sizeof (DATE));
    date->day = d;
    date->month = m;
    date->year = y;
    return date;
}

int main()
{
    DATE *now = createDate (10, 12, 2014);
    printdate (now);

    return 0;
}
```

Structure de structure I

- Champ dont le type est une **structure**
- On peut donc **imbriquer les structures**

```
struct person {  
    char *firstname;  
    char *lastname;  
    struct date birthdate;  
};
```

- Accès enchainé pour atteindre un champ imbriquée

```
struct person p;  
  
p.firstname = "Sébastien";  
p.birthdate.year = 1984;
```

Structure de structure II

3003	'\0'	
3002	'E'	
3001	'O'	
3000	'D'	
	...	
2004	'\0'	
2003	'N'	
2002	'H'	
2001	'O'	
2000	'J'	
	...	
1032	1975	(p.birthdate.year)
1024	3	(p.birthdate.month)
1016	17	(p.birthdate.day)
1008	3000	(p.lastname)
p : 1000	2000	(p.firstname)

```
struct person p;  
  
p.firstname = "John";  
p.lastname = "Doe";  
  
p.birthdate.day = 17;  
p.birthdate.month = 3;  
p.birthdate.year = 1975;
```

Tableau de structures

- Tableau de structure sur la pile...

```
struct person addressbook [5];
```

- ...ou en mémoire dynamique

```
struct person *addressbook =  
(struct person*) malloc (5 * sizeof (struct person));
```

- Accès aux champs avec l'opérateur d'accès (.)

```
1 addressbook [0].firstname = "John";
```