

BỘ CÔNG THƯƠNG
ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH



Bài giảng

NGÔN NGỮ LẬP TRÌNH C
THE C PROGRAMMING LANGUAGE

Lecturer : Le Ngoc Tran, PhD

Email : lengoctran@iuh.edu.vn

CHƯƠNG 7

CHUỖI

7.1. Giới thiệu chuỗi

- ❑ Chuỗi trong ngôn ngữ lập trình C thực chất là mảng một chiều của các ký tự mà kết thúc bởi một ký tự **null '\0'**
- ❑ Chuỗi ký tự kết thúc bởi một ký tự **'\0' (null)**
 - ❖ Độ dài chuỗi = kích thước mảng - 1
- ❑ Kiểu **char** chỉ chứa được một ký tự. Để lưu trữ một chuỗi (nhiều ký tự) ta sử dụng mảng (một chiều) các ký tự
- ❑ Ví dụ:
 - ❖ **char** hoten[30]; // Dài 29 ký tự

7.2. Khai báo chuỗi

- ❑ Để khai báo một chuỗi ta khai báo mảng ký tự 1 chiều.

char <tên chuỗi> [<số ký tự +1>]

- ❑ ví dụ:

❖ char **str**[20]; *//tạo ra 1 chuỗi có tên str với kích thước tối đa 20 ký tự.*

- ❑ Để khai báo một mảng chuỗi thì ta khai báo một mảng ký tự 2 chiều.

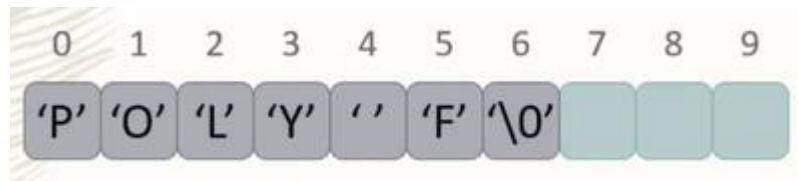
- ❑ Ví dụ:

❖ char **arrStr**[10][25]; *//tạo ra 1 mảng gồm 10 chuỗi và mỗi chuỗi có tối đa 25 ký tự.*

7.3. Khởi tạo

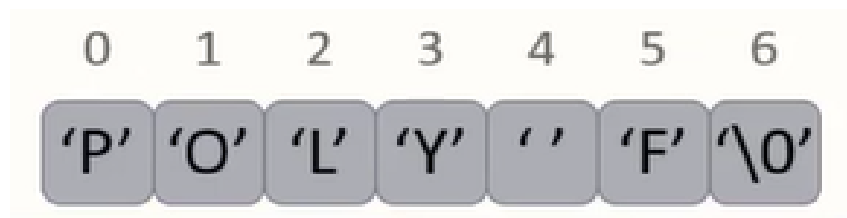
□ Độ dài cụ thể

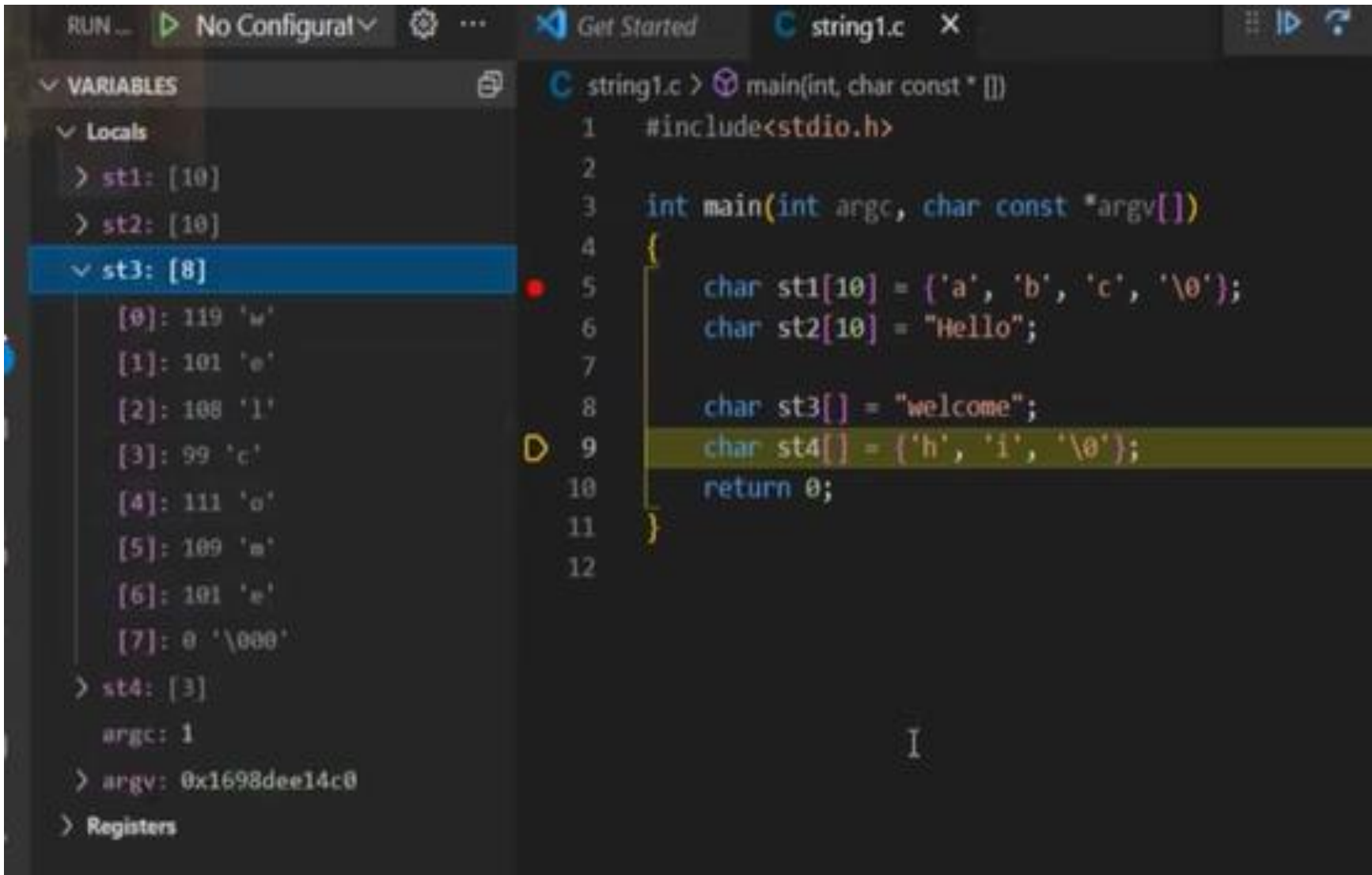
```
char s[10] = { 'P', 'O', 'L', 'Y', ' ', 'F', '\\0' };  
char s[10] = "POLY F"; // Tự động thêm '\\0'
```



□ Tự xác định độ dài

```
char s[] = { 'P', 'O', 'L', 'Y', ' ', 'F', '\\0' };  
char s[] = "POLY F"; // Tự động thêm '\\0'
```





The screenshot shows a C program named `string1.c` in Visual Studio Code. The program defines four character arrays: `st1` (size 10, containing 'a', 'b', 'c', and a null terminator), `st2` (size 10, containing 'Hello'), `st3` (size 8, containing 'welcome'), and `st4` (size 3, containing 'hi'). The `main` function returns 0. The left sidebar shows the 'VARIABLES' pane with 'Locals' expanded. `st3` is selected, showing its memory layout: `[0]: 119 'w'`, `[1]: 101 'e'`, `[2]: 108 'l'`, `[3]: 99 'c'`, `[4]: 111 'o'`, `[5]: 109 'm'`, `[6]: 101 'e'`, and `[7]: 0 '\000'`. Other variables shown are `st4: [3]`, `argc: 1`, and `argv: 0x1698dee14c0`.

```
string1.c > main(int, char const * [])
1  #include<stdio.h>
2
3  int main(int argc, char const *argv[])
4  {
5      char st1[10] = {'a', 'b', 'c', '\0'};
6      char st2[10] = "Hello";
7
8      char st3[] = "welcome";
9      char st4[] = {'h', 'i', '\0'};
10     return 0;
11 }
12
```

VARIABLES

Locals

- > st1: [10]
- > st2: [10]
- ▼ st3: [8]
 - [0]: 119 'w'
 - [1]: 101 'e'
 - [2]: 108 'l'
 - [3]: 99 'c'
 - [4]: 111 'o'
 - [5]: 109 'm'
 - [6]: 101 'e'
 - [7]: 0 '\000'
- > st4: [3]
- argc: 1
- > argv: 0x1698dee14c0
- > Registers

7.3. Xuất chuỗi

- ❑ Sử dụng hàm printf với đặc tả “%s”

```
char monhoc[50] = "Lap trinh C";  
printf("%s", monhoc);    // Không xuống dòng
```

```
Lap trinh C
```

- ❑ Sử dụng hàm puts

```
char monhoc[50] = "Lap trinh C";  
puts(monhoc);    // Tự động xuống dòng  
⇔ printf("%s\n", monhoc);
```

```
Lap trinh C
```

```
3 int main(int argc, char const *argv[])
4 {
5     char st1[10] = {'a', 'b', 'c', '\0'};
6     char st2[10] = "Hello";
7
8     char st3[] = "welcome";
9     char st4[] = {'h', 'i', '\0'};
10
11     printf("st1: %s \n", st1);
12     printf("st2: %s\n", st2);
13
14     puts(st3);
15     puts(st4);
16
17     return 0;
}
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\MyData\Demonstration\PolyCLang2021\Slide7> cd "c:\MyData"

st1: abc
st2: Hello
welcome
hi

7.4. Nhập chuỗi

❑ Sử dụng hàm scanf với đặt tả “%s”

- ❖ chỉ nhận các ký tự từ bàn phím đến khi gặp ký tự khoảng trắng hoặc ký tự xuống dòng.
- ❖ chuỗi nhận được không bao gồm ký tự khoảng trắng và xuống dòng.

❖ ví dụ:

```
char monhoc[50];  
printf("Nhap mot chuoi: ");  
scanf("%s", monhoc);  
printf("Chuoi nhan duoc la: %s", monhoc);
```

```
Nhap mot chuoi: Lap trinh C  
Chuoi nhan duoc la: Lap
```

Ví dụ

```
C string2.c > main(int, char const * [])
1  #include<stdio.h>
2
3  int main(int argc, char const *argv[])
4  {
5      char st1[10];
6
7      printf("Enter your name: ");
8      scanf("%s", st1);
9
10     printf("Your name is : %s \n", st1 );
11     puts(st1);
12
13
14     return 0;
15 }
16
```

```
PS C:\MyData\Demonstration\PolyCLang2021\Slide7> cd "c
Enter your name: Nguyen Ngoc Anh
Your name is : Nguyen
Nguyen
PS C:\MyData\Demonstration\PolyCLang2021\Slide7> |
```

7.4. Nhập chuỗi

❑ Sử dụng hàm **gets**

- ❖ Nhận các ký tự từ bàn phím đến khi gặp ký tự xuống dòng.
- ❖ Chuỗi nhận được là những gì người dung nhập (trừ ký tự xuống dòng).
- ❖ Ví dụ:

```
char monhoc[50];  
printf("Nhap mot chuoi: ");  
gets(monhoc);  
printf("Chuoi nhan duoc la: %s", monhoc);
```

```
Nhap mot chuoi: Lap trinh C  
Chuoi nhan duoc la: Lap trinh C
```

```
1  #include<stdio.h>
2
3  int main(int argc, char const *argv[])
4  {
5      char st1[10];
6
7      printf("Enter your name: ");
8      scanf("%s", st1);
9
10     printf("Your name is : %s \n", st1 );
11     puts(st1);
12
13     char st2[15];
14     printf("Enter your name: ");
15     fflush(stdin);
16     gets(st2);
17
18     printf("Your second name is: %s \n", st2);
19
20
21     return 0;
22 }
23
```

THƯ VIỆN STRING.H VÀ CÁC HÀM XỬ LÝ CHUỖI (STRING)

7.5. Hàm trong thư viện chuỗi

❑ Thư viện `<string.h>` định nghĩa các hàm cho phép thao tác với chuỗi như:

❖ `strlen`

❖ `strcpy`

❖ `strcat`

❖ `strcmp`

❖ `strrev`

❖ `strlwr`

❖ `strupr`

❖ `strstr`

7.5. Hàm trong thư viện chuỗi

❑ Hàm **strlen()**: trả về độ dài của chuỗi đã cho, không bao gồm ký tự NULL – ‘\0’.

❑ ví dụ:

```
char ch[20] = {'f', 'p', 'o', 'l', 'y', 'h', 'c', 'm', '\0'};  
printf("Do dai chuoi: %d",strlen(ch));
```

```
Do dai chuoi: 8
```

7.5. Hàm trong thư viện chuỗi

❑ Hàm **strcpy(target, source)**: sao chép chuỗi ký tự từ **source** tới target.

❑ ví dụ:

```
char source[20] = {'f', 'p', 'o', 'l', 'y', 'h', 'c', 'm', '\0'};  
char target[20];  
strcpy(target, source);  
printf("Chuoi target: %s", target);
```

```
Chuoi target: fpolyhcm
```


7.5. Hàm trong thư viện chuỗi

❑ Hàm **strcat()**: nối 2 chuỗi

❑ ví dụ:

```
char ch [10] = { 'H' , 'e' , 'l' , 'l' , 'o' , '\0' };  
char ch2 [10] = { 'C' , '\0' };  
strcat (ch, ch2);  
printf ( "Chuoi sau khi da ket noi la: %s", ch);
```

```
Chuoi sau khi da ket noi la: HelloC
```

```
C build-in-func1.c > main(int, char const * [])
1  #include<stdio.h>
2  #include<string.h>
3
4  int main(int argc, char const *argv[])
5  {
6      char st1[20] = "Hello World";
7      char st2[20];
8
9      printf("Size of st1: %d \n", strlen(st1));
10
11     strcpy(st2, st1);
12
13     printf("Value of st2: %s \n", st2);
14
15     strcat(st1, st2);
16
17     printf("Value of st1: %s \n", st1);
18     printf("Value of st2: %s \n", st2);
19
20     return 0;
21 }
22
```

```
PS C:\MyData\Demonstration\PolyCLang2021\Slide7> cd "
Size of st1: 11
Value of st2: Hello World
Value of st1: Hello WorldHello World
Value of st2: Hello World
PS C:\MyData\Demonstration\PolyCLang2021\Slide7> █
```

7.5. Hàm trong thư viện chuỗi

- ❑ Hàm **strcmp()**: so sánh 2 chuỗi (phân biệt hoa thường).
- ❑ Nếu giá trị trả về:
 - ❖ $=0$ thì 2 chuỗi bằng nhau
 - ❖ >0 thì chuỗi str1 lớn hơn chuỗi str2
 - ❖ <0 thì chuỗi str1 nhỏ hơn chuỗi str2

Hàm strcmp():

```
printf("Nhap chuoi 1: ");
gets(str1);
printf("Nhap chuoi 2: ");
gets(str2);

if (strcmp(str1, str2) == 0) {
    printf("2 chuoi bang nhau.");
} else if (strcmp(str1, str2) > 0) {
    printf("Chuoi 1 lon hon chuoi 2");
} else {
    printf("Chuoi 1 nho hon chuoi 2");
}
```

```
C string-cmp.c > main(int, char const * [])
1  #include<stdio.h>
2  #include<string.h>
3
4  int main(int argc, char const *argv[])
5  {
6      char st1[20] = "AnhNN";
7      char st2[20] = "AnhNN";
8
9      int result = strcmp(st1, st2);
10     printf("Result is %d \n", result);
11
12     if (result == 0){
13         printf("Equals \n");
14     }else if (result < 0){
15         printf("st1 < st2");
16     }else {
17         printf("st1 > st2");
18     }
19 }
```

```
PS C:\MyData\Demonst
Result is 0
Equals
```

7.5. Hàm trong thư viện chuỗi

- ❑ Hàm **strrev()**: trả về chuỗi đảo ngược chuỗi đã cho
- ❑ Ví dụ:

```
char str[20];  
printf("Nhap chuoi: ");  
gets(str);
```

```
Chuoi da nhap: ipolyhcm  
Chuoi dao nguoc: mchylopf
```

```
printf("Chuoi da nhap: %s", str);  
printf("\nChuoi dao nguoc: %s", strrev(str));
```

```
C string-reverse.c > ...
1  #include<stdio.h>
2  #include<string.h>
3
4  int main(int argc, char const *argv[])
5  {
6      char st[20];
7
8      printf("Enter your name: ");
9      gets(st);
10
11     printf("Your name: %s \n", st);
12     printf("Your reversed name: %s \n", strrev(st));
13
14     return 0;
15 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/powershell>

PS C:\MyData\Demonstration\PolyCLang2021\Slide7> cd "c:\MyData\Demo
Enter your name: Nguyen Ngoc Anh
Your name: Nguyen Ngoc Anh
Your reversed name: hñA cogN neyugN
PS C:\MyData\Demonstration\PolyCLang2021\Slide7>

7.5. Hàm trong thư viện chuỗi

- ❑ Hàm **strlwr()**: trả về chuỗi thường từ chuỗi đã cho
- ❑ Ví dụ:

```
char str[20];  
printf("Nhap chuoi: ");  
gets(str);
```

```
printf("Chuoi da nhap la: %s",str);  
printf("\nChuoi chu thuong la: %s",strlwr(str));
```

```
Nhap chuoi: FPOLYHCM  
Chuoi da nhap: FPOLYHCM  
Chuoi chu thuong la: fpolyhcm
```


7.5. Hàm trong thư viện chuỗi

- ❑ Hàm **strupr()**: trả về chuỗi chữ HOA từ chuỗi đã cho
- ❑ Ví dụ:

```
char str[20];  
printf("Nhap chuoi: ");  
gets(str);  
  
printf("Chuoi da nhap la: %s",str);  
printf("\nChuoi chu HOA la: %s",strupr(str));
```

```
Nhap chuoi: fpolyhcm  
Chuoi da nhap: fpolyhcm  
Chuoi chu HOA la: FPOLYHCM
```

```
C string-reverse.c > main(int, char const * [])
1  #include<stdio.h>
2  #include<string.h>
3
4  int main(int argc, char const *argv[])
5  {
6      char st[20];
7
8      printf("Enter your name: ");
9      gets(st);
10
11     printf("Your name: %s \n", st);
12     printf("Your reversed name: %s \n", strrev(st));
13
14     printf("Your UPPER CASE name: %s \n",strupr(st));
15     printf("Your lower case name: %s \n",strlwr(st));
16 }
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\MyData\Demonstration\PolyCLang2021\Slide7> cd "c:\MyData\Demonstration\PolyCLang2021\Slide7"

Enter your name: Nguyen Ngoc Anh
Your name: Nguyen Ngoc Anh
Your reversed name: hNA cogN neyugN
Your UPPER CASE name: HNA COGN NEYUGN
Your lower case name: hna cogn neyugn
PS C:\MyData\Demonstration\PolyCLang2021\Slide7>

7.5. Hàm trong thư viện chuỗi

❑ Hàm **strstr()**: tìm chuỗi con từ vị trí khớp đầu tiên cho đến lần khớp cuối cùng.

❑ Ví dụ:

```
char str1[30] = "Learning a C is awesome";  
char str2[15] = "C";
```

```
if(strstr(str1, str2) != NULL)  
    printf("tim thay");
```

7.6. Bài tập chuỗi

- Bài tập 1: Đếm nguyên âm và phụ âm của chuỗi
 - ❖ Nhập vào chuỗi
 - ❖ Hiển thị số lượng nguyên âm và phụ âm
 - ❖ Nguyên âm: a, e, i, o, u

7.6. Bài tập chuỗi

□ Bài tập 2: Kiểm tra username và password

- ❖ Nhập vào username và password
- ❖ Hiển thị đăng nhập thành công hay không so với username và password đặt trước

7.6. Bài tập chuỗi

□ Bài tập 3: Sắp xếp chuỗi

- ❖ Nhập vào 5 chuỗi

- ❖ Sắp xếp và hiển thị các chuỗi sau khi sắp xếp

