C Programming Basic – week 1

Giới thiệu

- Lập trình C trên môi trường Linux
- Thực hanh cấu trúc dữ liệu & giải thuật
- Trình biên dịch: gcc
- Bộ soạn thảo: Emacs, K-Developper.

Cú pháp gcc

- Tham số:
  - -Wall : bật các cảnh báo
  - -c: tạo tệp object
  - -o: tên tập đầu ra
  - -g: thông tin debug
  - -l: thư viện

```
gcc –Wall hello.c –o runhello
./runhello
```

Nội dung

- Chủ đề:
  - Mảng, chuỗi, con trỏ
  - Đọc/ghi tệp văn bản
  - Bài tập

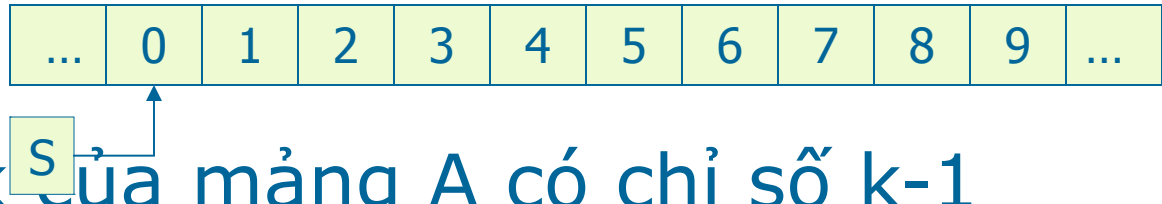
Mảng

- Khối các biến cùng kiểu
- Chứa bất kỳ kiểu dữ liệu nào
  - VD `int A[10]` là mảng chứa 10 phần tử kiểu int
- Ví dụ:
  - Danh sách điểm của sinh viên
  - Chuỗi số nhập vào bởi người dùng
  - Vector
  - Ma trận

Mảng trong bộ nhớ

- Chuỗi các biến có kiểu xác định
- Biến mảng chứa địa chỉ (của phần tử đầu tiên) của mảng
- V:

```
double S[10];
```



- Phần tử thứ k của mảng A có chỉ số k-1 **(0-based)**

VD:

```
#include <stdio.h>

int main(void)
{
   int i, A[10];

   printf("please enter 10 numbers:\n");
   for(i=0; i<10; i++)
      scanf("%d", &A[i]);

   printf("numbers in reversed order:\n");
   for(i=9; i>=0; i--)
      printf("%d\n", A[i]);

   return 0;
}
```

Exercise 1.1

- Viết chương trình nhận một dòng từ người dùng và hiển thị số lần xuất hiện của mỗi kí tự trong dòng đó

The output for the input line: "hello, world!"

The letter 'd' appears 1 time(s).
The letter 'e' appears 1 time(s).
The letter 'h' appears 1 time(s).
The letter 'l' appears 3 time(s).
The letter 'o' appears 2 time(s).
The letter 'r' appears 1 time(s).
The letter 'w' appears 1 time(s).

Giả sử tất cả các kí tự đều viết thường!

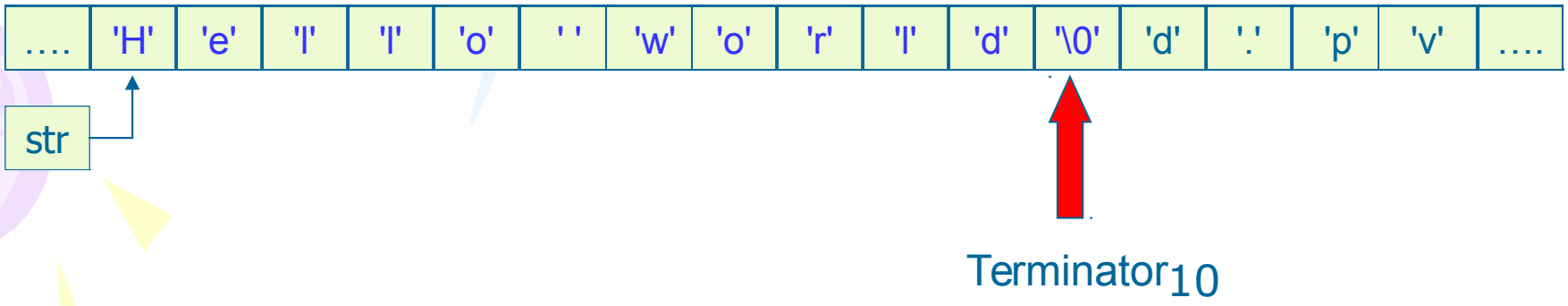
Exercise 1.2

- Cài đặt một hàm nhận vào hai số nguyên, trả về 0 nếu chúng bằng nhau, trả về 1 nếu ngược lại
- Viết chương trình sử dụng hàm vừa tạo

Chuỗi

- Một mảng các kí tự
- Sử dụng để lưu trữ văn bản
- Khởi tạo:

```
char str[] = "Text";
```



Chuỗi (2)

- Cần một mảng độ dài N+1 để lưu một chuỗi có N kí tự
- Kí tự '\0' dùng để kết thúc chuỗi

```
char str[] = {'b', 'l', 'a', 'b', 'l',
'a', '\0'};
```

Các hàm xử lý chuỗi và kí tự

- getchar()
  - c = getchar()
- scanf
  - scanf("%s", str);
- gets()
  - gets(str);

Các hàm xử lý chuỗi và kí tự (2)

- **`strlen(const char s[])`**
  trả về độ dài của chuỗi s
- **`strcmp(const char s1[],`**
  `const char s2[])`
  so sánh chuỗi s1 và s2
- **`strcpy(char s1[],`**
  `const char s2[])`
  sao chép chuỗi s2 vào chuỗi s1

Exercise 1.3

- Viết một hàm
  - Nhận vào một chuỗi và hai kí tự
  - Thay thế tất cả các chỗ trong chuỗi kí tự đầu tiên bằng kí tự thứ hai
- Viết chương trình sử dụng hàm nói trên
  - Chương trình đọc vào chuỗi và hai kí tự từ người dùng và in ra kết quả sau khi thay thế
- VD:
  - Đầu vào: “papa”, ‘p’, ‘m’
  - Đầu ra: “mama”

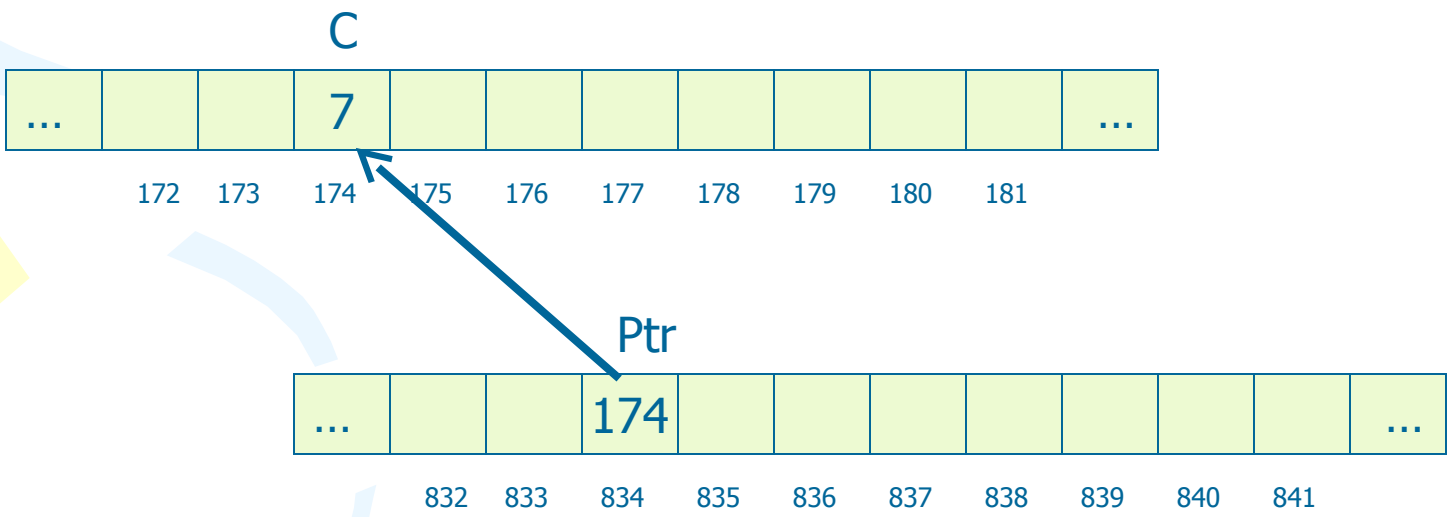
Con trỏ

```
type *variable_name;
```

- Con trỏ được khai báo bằng dấu * trước tên biến
- Con trỏ là biến có giá trị là một địa chỉ trong bộ nhớ
- Địa chỉ phải là địa chỉ của một biến hoặc một mảng

Con trỏ (2)

- Ở đây ptr được gọi là *trỏ* tới địa chỉ của biến c



Con trỏ (3)

```
int n;
int *iptr; /* Declare P as a pointer to int */
n = 7;
iptr = &n;

printf("%d", *iptr); /* Prints out '7'*/
*iptr = 177;
printf("%d", n); /* Prints out '177' */
iptr = 177; /* This is unadvisable!! */
```

Exercises 1.4

Viết một hàm nhận vào một số thực và trả về phần nguyên và phần thập phân của nó

Viết chương trình sử dụng hàm trên

Exercise 1.5

- Viết hàm với nguyên mẫu sau:

```
void replace_char(char *str,
                  char c1,
                  char c2);
```

- **thay thế c1** bởi **c2** trong chuỗi **str**.

  Không sử dụng toán tử []!

- Viết chương trình sử dụng hàm trên

Tham số dòng lệnh

- Là tham số của hàm **`main`**
  - Main() về cơ bản là một hàm
  - Có thể nhận tham số giống như các hàm khác
  - Lời gọi hàm xuất phát từ hệ điều hành hoặc các chương trình khác

Nguyên mẫu hàm 'main'

```
int main(int argc, char* argv[])
```

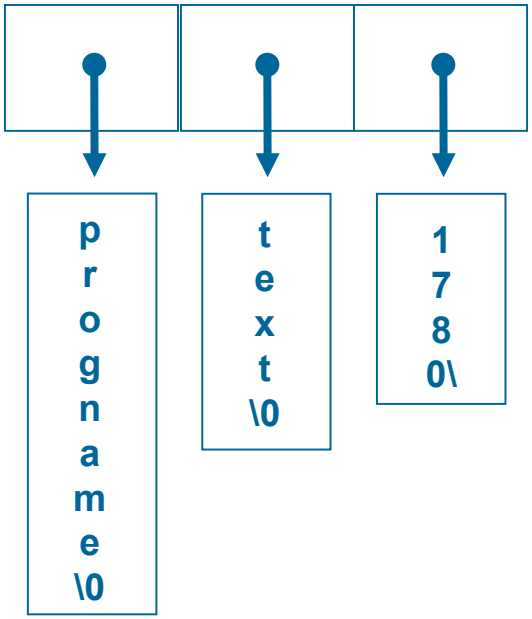
- Nếu muốn **main** nhận tham số dòng lệnh cần khai báo
  - **argc** là số tham số
  - **argv** là một mảng các chuỗi chứa giá trị các tham số dưới dạng chuỗi
- Tham số đầu tiên luôn luôn là tên chương trình

Nguyên mẫu hàm 'main' (2)

```
int main(int argc, char* argv[])
```

argc : 3

argv :



Exercise 1.6

- Viết chương trình nhận hai số thực dưới dạng tham số dòng lệnh là chiều dài và chiều rộng của hình chữ nhật
- Chương trình in ra chu vi và diện tích của hình chữ nhật tương ứng

Homework 1

- Viết chương trình nhận tham số dòng lệnh tính e^x với cú pháp:
- E 50

Xử lý tệp

- C giao tiếp với tệp dựa trên con trỏ tệp

- Con trỏ tệp:
  - Tham chiếu đến một tệp trên đĩa
  - sử dụng bởi một stream để thực thi vào ra với tệp

- FILE *fptr;

Các thao tác chính

- Mở tệp
- Đọc từ tệp vào chương trình
- Viết từ chương trình ra tệp
- Đóng tệp

Mở tệp

- FILE *fopen(const char *filename, const char *mode);

```
FILE *fptr;
if ((fptr = fopen("test.txt", "r")) ==
  NULL){
    printf("Cannot open test.txt file.\n");
    exit(1);
}
```

Mở tệp (2)

- filename: tên của tệp
  - Có thể là một hằng chuỗi **"data.txt"**
  - Có thể chứa đường dẫn đầy đủ của tệp **"/root/hedspi/CProgrammingBasic/Lab1/data.txt"**
  - Có thể là một mảng kí tự

```
char file_name[] = "junk.txt";
```

- ***NOTE:*** *Nếu không có đường dẫn, đường dẫn mặc định là cùng thư mục với chương trình*

Các tùy chọn

Tùy chọn	Mô tả
"r"	mở tệp để đọc
"w"	tạo ra tệp để ghi
"a"	mở tệp có sẵn để thêm vào
"r+"	mở tệp có sẵn để đọc hoặc ghi
"w+"	tạo ra một tệp để đọc hoặc ghi
"a+"	mở tệp có sẵn hoặc tạo tệp mới để thêm vào

Các tùy chọn nhị phân

mode	Mô tả
"rb"	mở tệp nhị phân để đọc
"wb"	tạo tệp nhị phân để ghi
"ab"	mở tệp nhị phân đã có để thêm vào
"r+b"	mở tệp nhị phân đã có để đọc hoặc ghi
"w+b"	tạo tệp nhị phân để đọc hoặc ghi
"a+b"	mở tệp có sẵn hoặc tạo tệp mới để thêm vào

Đóng tệp

- Hàm fclose dùng để bỏ liên kết từ một con trỏ tệp tới tệp

- int fclose(FILE *stream);

VD: Mở và đóng tệp

```
1:  /* Opening and closing a file */
2:  #include <stdio.h>
3:
4:  enum {SUCCESS, FAIL};
5:
6:  main(void)
7:  {
8:     FILE *fptr;
9:     char filename[]= "haiku.txt";
10:    int reval = SUCCESS;
11:
12:    if ((fptr = fopen(filename, "r")) == NULL){
13:       printf("Cannot open %s.\n", filename);
14:       reval = FAIL;
15:    } else {
16:       printf("The value of fptr: 0x%p\n", fptr);
17:       printf("Ready to close the file.");
18:       fclose(fptr);
19:    }
20:
21:    return reval;
22: }
```

Đọc/ghi tệp văn bản

- Các cách đọc/ghi tệp trong C
  - Đọc/ghi từng kí tự
  - **Đọc/ghi một dòng**
  - Đọc/ghi một khối kí tự

Đọc/ghi dựa trên kí tự

- **Đọc/ghi từng kí tự**

- Đọc và ghi kí tự
  - fgetc() và fputc()

- int fgetc(FILE *stream);

- int fputc(int c , FILE *stream);

Exercise 1.7

- Tạo tệp văn bản lab1.txt với nội dung bất kỳ
- Viết chương trình đọc từ tệp lab1.txt từng kí tự và ghi ra tệp lab1w.txt

Homework 2

- Viết chương trình có tên *mycp* để sao chép nội dung từ tệp nguồn sang tệp đích với cú pháp:

- mycp a1.txt a2.txt

Exercise 1.8

- Viết chương trình đọc các câu của tệp văn bản mỗi kí tự một lần
- Kí tự viết thường được chuyển thành viết hoa và ngược lại. Ghi câu mới vào một tệp văn bản khác
- Lưu ý giữ nguyên giá trị các dấu câu và số.

Đọc/ghi dòng

- Hai hàm: fgets() và fputs()
- char *fgets(char *s, int n, FILE *stream);
  - s là một chuỗi
  - n số kí tự tối đa của chuỗi
- fgets() đọc được tối đa n-1 kí tự cho tới khi gặp dấu xuống dòng hoặc EOF và chèn thêm kí tự '\0' vào cuối

Đọc/ghi dòng (2)

- int fputs(const char *s, FILE *stream);
- s: chuỗi cần ghi
- kết quả trả về
  - 0 nếu thành công
  - Khác 0 nếu thất bại

Exercise 1.9

- Yêu cầu tương tự Exercise 1.7 nhưng đọc/ghi từng dòng

Exercise 1.10

- Viết chương trình có tên *mycat* đọc nội dung tệp và in ra màn hình. Chương trình có thể nhận 1 hoặc 2 tham số dòng lệnh
- cat <filename> : hiển thị toàn bộ nội dung
- cat <filename> -p : hiển thị nội dung theo trang

Đọc/ghi theo khuôn dạng

- int fscanf(FILE *stream, const char *format, ...);
  - Hoạt động tương tự scanf nhưng đọc vào từ tệp
- int fprintf(FILE *stream, const char *format, ...);
  - Hoạt động tương tự như printf nhưng ghi ra tệp

Homework 3

- Viết chương trình đọc một tệp văn bản. Thêm số thứ tự của dòng vào đầu và in ra màn hình. Chương trình nhận tên tệp là tham số dòng lệnh

- Nội dung tệp:

  This is sample file.

  Hello!

- In ra màn hình:

  1 This is sample file.

  2 Hello!

Homework 4

Viết chương trình so sánh nội dung hai tệp với tên tệp là tham số dòng lệnh. Chương trình in ra những dòng mà hai tệp khác nhau