ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LẬP TRÌNH C CƠ BẢN

Kiểu dữ liệu cơ bản, vào ra file

Ngô Văn Linh

Giới thiệu về môn học

- Thực hành lập trình theo các chủ đề về cấu trúc dữ liệu và giải thuật
  - Mức độ: cơ sở
  - Cài đặt các cấu trúc dữ liệu – thuật toán
  - Ứng dụng vào giải quyết một số bài toán thực tế.
- Ngôn ngữ lập trình: C

Giới thiệu về môn học

- Môi trường khuyến nghị:
  - Hệ điều hành UNIX
- Chương trình dịch: gcc
- Chương trình soạn thảo mã nguồn: Emacs, K-Developper.

Đánh giá kết quả môn học

- Điểm quá trình: 30%
  - Chấm điểm bài tập về nhà hàng tuần, kết hợp với chất lượng hoàn thành thực hành trên lớp.
  - Mỗi tuần: 4, 5 sinh viên.
- Cuối kỳ: 70%
  - Thi lập trình trên máy tính.
  - Điểm đánh giá dựa trên kết quả chạy chương trình, không đánh giá dựa trên mã nguồn.

Cú pháp dịch chương trình bằng gcc

- Các tham số:
  - -Wall : bật tất cả các cảnh báo
  - -c: tạo tập tin object
  - -o: tạo tập tin chương trình
  - -g: thêm thông tin gỡ rối
  - -l: sử dụng kèm thư viện

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

Chủ đề

- Ôn tập về các kiểu dữ liệu:
  - mảng,
  - xâu ký tự,
  - con trỏ.
- Xây dựng chương trình với đối số dòng lệnh
- Thao tác với tập tin văn bản
  - Đọc ghi tập tin theo từng ký tự
  - Đọc ghi tập tin theo từng dòng
  - Đọc ghi tập tin theo đặc tả định dạng
- Các bài tập lập trình

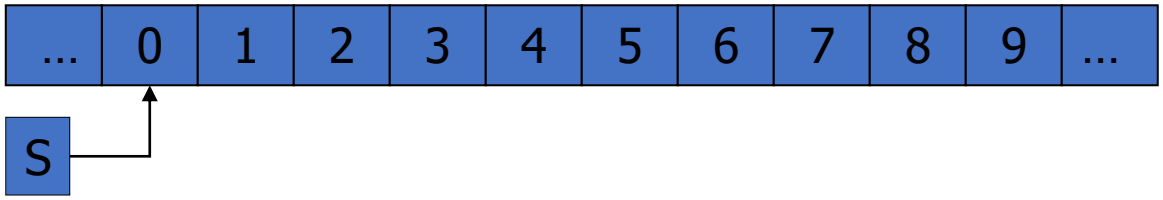
I. Mảng

- Khối liên tục các biến cùng kiểu dữ liệu, cùng tên
- Mảng có thể khai báo với bất cứ kiểu dữ liệu nào
  - VD: `int A[10];` là khai báo mảng 10 số nguyên.
- Các ví dụ về sử dụng mảng
  - Danh sách điểm của các sinh viên
  - Dãy số nhập từ người dùng
  - Véc tơ
  - Ma trận

Mảng trong bộ nhớ trong

- Dãy liên tục các biến cùng kiểu, nằm kế tiếp nhau
- Tên mảng chứa địa chỉ bộ nhớ của ô nhớ đầu tiên
- Ví dụ:

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



- Phần tử thứ k của mảng A truy cập qua chỉ số: A[k-1] **(bắt đầu từ 0)**

Ví dụ - hiển thị mảng theo chiều ngược

```
#include <stdio.h>

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

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

   printf("numbers in reversed order:\n");
   for(i=9; i>=0; i--)
      printf("%3.4f\n", A[i]);

   return 0;
}
```

Bài tập : Tìm phần tử đặc biệt trong mảng

- Viết tiếp chương trình ví dụ trên để hiển thị phần tử mảng có giá trị gần nhất với giá trị trung bình của tất cả các phần tử mảng

- Lưu ý: Phải sử dụng HẰNG CONSTANT trong khai báo kích thước mảng.

Hướng dẫn – mã giả

```
avg = average(a, SIZE); // tính giá trị trung bình các phần tử mảng
mindiff = 1000000;
for (i=0; i<SIZE; i++){
      diff = fabs(a[i] - avg);
      if (diff <mindiff) {
         mindiff = diff;
         medianindex = i;
      }
}
```

➔ return a[medianindex]

Bài tập

- Viết chương trình nhập vào một xâu ký tự và hiển thị số lần xuất hiện của mỗi ký tự có trong xâu.

Kết quả hiển thị với xâu đầu vào: "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 các ký tự trong xâu đều là chữ cái thường!

Đáp án (phiên bản 1)

```
#define ALPHABET_LEN 26

int main(void)
{
    int i = 0,
   count[ALPHABET_LEN] = {0};
   char c = '\0';

   printf("Please enter a line of text: \n");

   /* Read in letter by letter and update the count array */
    c = getchar();
   while (c != '\n'  &&  c >= 0)
   {
       if (c <= 'z'  &&  c >= 'a')
       ++count[c - 'a'];
       if (c <= 'Z'  &&  c >= 'A')
       ++count[c - 'A'];
       c = getchar();
   }
```

Đáp án (phiên bản 1)

```
    for (i = 0; i < ALPHABET_LEN ; ++i)     {
        if (count[i] > 0)
        printf("The letter '%c' appears %d time(s).\n", 'a'
        + i, count[i]);
    }
    return 0;
}
```

Đáp án (phiên bản 2)

```
#include <stdio.h>
#include <string.h>
#define ALPHABET_LEN 26
int main(){
   int i = 0,
   count[ALPHABET_LEN] = {0};
    char s[20], c = '\0';
    printf("Please enter a line of text: \n"); gets(s);
    for (i=0; i<strlen(s);i++) {
         c = s[i];
        if (c <= 'z'  &&  c >= 'a')    ++count[c - 'a'];
        if (c <= 'Z'  &&  c >= 'A')    ++count[c - 'A'];
    }
    for (i = 0; i < ALPHABET_LEN ; ++i) {
         if (count[i] > 0)
         printf("The letter '%c' appears %d time(s).\n", 'a' + i,
         count[i]);
   }
return 0;
}
```

Bài tập: Hàm có mảng là đối số

- Viết một hàm nhận đối số là hai mảng số nguyên, có cùng số phần tử. Hàm trả về 1 nếu hai mảng có nội dung trùng nhau và 0 cho các trường hợp còn lại.
- Kiểm tra hoạt động của hàm qua một chương trình.
- Hoàn thành tại nhà: Hàm trả về -1 nếu 2 mảng có nội dung đối xứng (mảng này ngược với mảng kia)

Đáp án

```
#include <stdio.h>

#define SIZE 5

int compare_arrays(int arr1[], int arr2[], int size)
{
   int i = 0;

   for (i = 0; i < size; ++i)
   {
       if (arr1[i] != arr2[i])
           return 0;
   }

    /* if we got here, both arrays are identical */
    return 1;
}
```

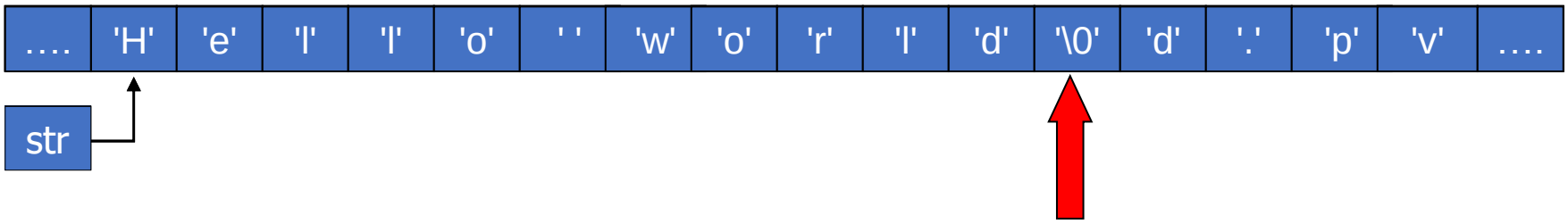
Đáp án

```
int main(void)
{
   int input1[SIZE], input2[SIZE], i;

   printf("Please enter a list of %d integers:\n", SIZE);
   for (i = 0; i < SIZE; ++i)   scanf("%d", &input1[i]);

   printf("Please enter another list of %d integers:\n",
   SIZE);
   for (i = 0; i < SIZE; ++i) scanf("%d", &input2[i]);

   if (compare_arrays(input1, input2, SIZE) == 1)
       printf("Both lists are identical!\n");
   else
       printf("The lists are not identical...\n");

   return 0;
}
```

II. Xâu ký tự trong C

- Mảng các ký tự.
- Kiểu dữ liệu chính để lưu trữ văn bản.
- Có thể khởi tạo khi khai báo:

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



Xâu ký tự

- In order to hold a string of N characters we need an array of length N + 1
- So the previous initialization is equivalent to

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

Các hàm nhập xâu ký tự và ký tự

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

Các hàm xử lý xâu ký tự

- `strlen(const char s[])`
  - trả về số ký tự của xâu s (không tính ký tự NULL)
- `strcmp(const char s1[], const char s2[])`
  - so sánh s1 với s2
- `strcpy(char s1[], const char s2[])`
  - sao chép nội dung s2 vào s1
- `strcat(char s1[], char s2[])`
  - nối s2 vào s1, sau đó lưu kết quả vào s1

Bài tập

- Viết hàm :
  - Có tham số là một xâu ký tự và hai ký tự
  - Hàm sẽ duyệt xâu và thay thế tất cả các ký tự thứ nhất trong xâu bằng ký tự thứ hai.
- Viết chương trình để kiểm tra hàm nói trên:
  - Đọc một xâu không chứa ký tự trắng và hai ký tự, sau đó gọi hàm với các đối số trên và in ra kết quả.
- Ví dụ
  - Đầu vào: “papa”, ‘p’, ‘m’
  - Kết quả: “mama”

Đáp án (hàm)

```
void replace(char str[], char replace_what,
                         char replace_with)
{
    int i;

    for (i = 0; str[i] != '\0'; ++i)
    {
        if (str[i] == replace_what)
            str[i] = replace_with;
    }
}
```

\

Đáp án (chương trình)

```c
#define STRING_LEN 100

int main(void)
{
    char str[STRING_LEN + 1];
    char replace_what, replace_with, tmp;

    printf("Please enter a string (no spaces)\n");
    scanf("%100s", str);

    printf("Letter to replace: ");
    scanf(" %c", &replace_what);
       do {tmp=getchar();} while (tmp!='\n');

    printf("Letter to replace with: ");
    scanf(" %c", &replace_with);

    replace(str, replace_what, replace_with);
    printf("The result: %s\n", str);
    return 0;
}
```

Bài tập: Tách từ

- Viết chương trình đọc một xâu ký tự biểu diễn một câu từ người dùng. Sau đó chương trình hiển thị mỗi từ trong câu trên một dòng. Một từ là một dãy các ký tự liên tiếp không chứa ký tự trắng.
- Ví dụ:
  - Đầu vào: « The house nextdoor is   very old. »
  - Kết quả:
    - The
    - house
    - ….

Bài tập

- Viết chương trình yêu cầu người dùng nhập số lượng sinh viên trong một lớp học, sau đó nhập tên đầy đủ bằng tiếng Việt của mỗi sinh viên. Hiển thị danh sách sinh viên sắp xếp theo tên sinh viên. Ví dụ:
  - Nguyen Bao Anh
  - Tran Quang Binh
  - Vuong Quoc Binh
  - Dao Thi Ha
  - Ngo Anh Vu
- Nâng cao (không bắt buộc): Hiển thị số lượng lớn nhất các sinh viên cùng tên.

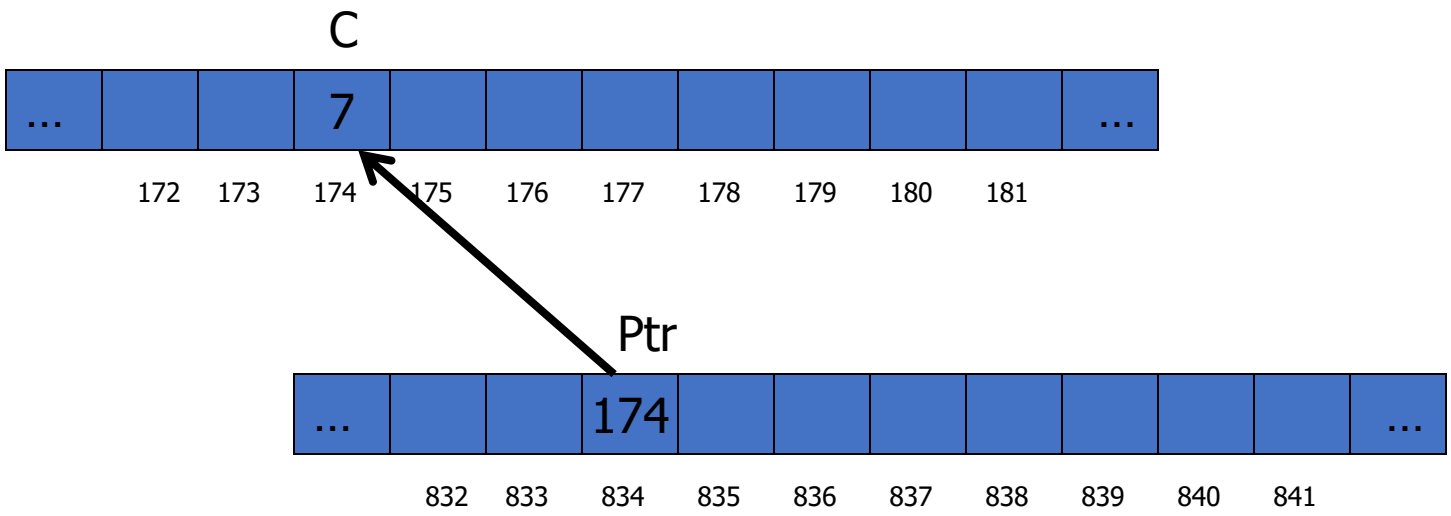
III. Con trỏ

- Con trỏ là biến dùng để lưu giá trị một địa chỉ bộ nhớ.
- Địa chỉ của biến hoặc mảng.
- Được khai báo với ký tự * trước tên.

```
Kiểu_dữ_liệu *tên_biến;
```

Con trỏ

- Con trỏ ptr được gọi là "trỏ" *point* tới biến c nếu giá trị của nó là địa chỉ của c



Toán tử tham chiếu (reference) and Toán tử điều hành gián tiếp (indirection)

```
int n;
int *iptr; /* khai báo P là một con trỏ
   kiểu int */
n = 7;
iptr = &n;

printf("%d", *iptr); /* Hiển thị '7'*/
*iptr = 177;
printf("%d", n); /* Hiển thị '177' */
iptr = 177; /* Phép gán không đúng!! */
```

Bài tập

Viết hàm nhận đối số là một số thực (double) và trả về phần nguyên và phần thập phân của số đó.

Viết chương trình minh họa hàm trên, với số thực được nhập từ người dùng.

Đáp án

```
void split(double num, int *int_part, double *frac_part)
{
        *int_part = (int)num;
        *frac_part = num - *int_part;
}

int main(void)
{
        double num, fraction;
        int integer;

        printf("Please enter a real number: ");
        scanf("%f", &num);

        split(num, &integer, &fraction);
        printf("The integer part is %d\n", integer);
        printf("The remaining fraction is %f\n", fraction);

        return 0;
}
```

Bài tập

- Write a function with the prototype:

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

- It replaces each appearance of **c1** by **c2** in the string **str**.
  Do not use the [] operator!
- Demonstrate your function with a program that uses it

Đáp án

```
void replace_char(char *str, char c1, char c2)
{
    if (str == NULL)
        return;

    while (*str != '\0')
    {
        if (*str == c1)
            *str = c2;

        ++str;
    }
}
```

Bài tập về nhà: Sinh câu văn tự động

- Viết chương trình có khả năng sinh ra các câu tự động sử dụng kỹ thuật lựa chọn dựa trên số ngẫu nhiên. Chương trình dùng bốn mảng xâu ký tự để lưu trữ **các mạo từ (article), danh từ (noun), động từ (verb), giới từ (preposition)**. Câu được tạo ra bằng cách lựa chọn ngẫu nhiên các phần tử trong các mảng trên và ghép lại theo thứ tự: **mạo từ, danh từ, động từ, mạo từ và danh từ**. Câu sinh ra cần bắt đầu với kỹ tự in hoa và kết thúc với dấu chấm. Chương trình cần sinh ra tối thiểu 10 câu.

- Ví dụ về các phần tử mảng
  - mạo từ: "the", "a", "one", "some" and "any";
  - danh từ: "boy", "girl", "dog", "town" and "car";
  - động từ: verbs "drove", "jumped", "ran", "walked" and "skipped";
  - giới từ: "to", "from", "over", "under" and "on".

IV. Chương trình với đối số dòng lệnh

- Các đối số dòng lệnh được định nghĩa trong hàm **`main`**
  - main bản thân là một hàm như các hàm khác
  - Nó có thể nhận các đối số truyền vào (từ dòng lệnh gõ bởi người dùng)
  - Vai trò của chương trình gọi nó (the calling function) trong trường hợp này là hệ điều hành, hoặc một chương trình khác

Nguyên mẫu của hàm main

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

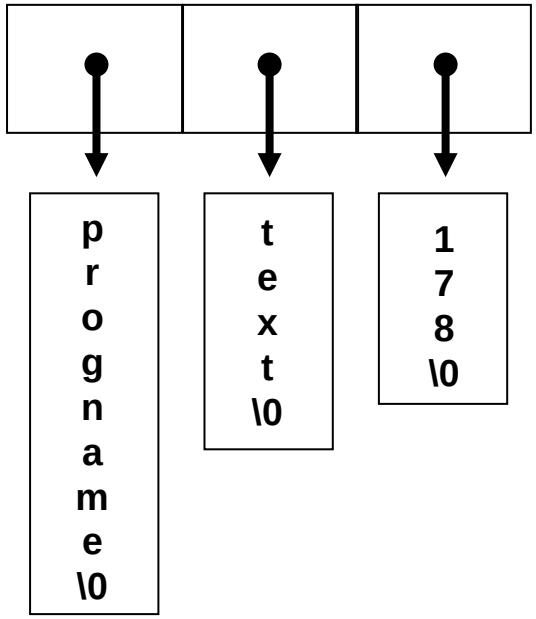
- Khi chúng ta muốn chương trình nhận các đối số tại dòng lệnh, cần định nghĩa hàm main như trên, với:
  - `argc` chứa số lượng các đối số
  - `argv` là một mảng các con trỏ kiểu char – mảng các xâu ký tự – nhận và lưu trữ các giá trị của các đối số dưới dạng dữ liệu văn bản.
- Đối số đầu tiên mặc định luôn là tên của chương trình.

Nguyên mẫu của hàm main

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

```
argc : 3
```

```
argv :
```



Các bước viết chương trình nhận đối số dòng lệnh

- Viết đúng cú pháp cho hàm **main**
- Kiểm tra giá trị **argc** để đảm bảo người dùng nhập đúng số đối số. Nếu sai, hiển thị thông báo cùng hướng dẫn về cú pháp sử dụng chương trình.
  - VD: Hello Hanoi ➔ 1 đối số thực sự, giá trị của argc nên là : 1+1 =2
- Lấy giá trị của các đối số từ mảng argv[], bắt đầu từ argv[1] và chuyển đổi sang đúng kiểu dữ liệu khi cần thiết.
  - Sử dụng các hàm atoi, atol, atof trong thư viện <stdlib.h>
- Xử lý, tính toán trên các đối số nói trên.

Bài tập

- Viết chương trình nhận hai số thực làm đối số dòng lệnh, đại diện cho chiều dài và chiều rộng của một hình chữ nhật.
- Chương trình dựa trên đó tính toán và in ra diện tích và chu vi của hình chữ nhật.

Đáp án

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    double width, height;
    if (argc != 3)
        printf("Wrong number of arguments!\n");
         printf("CORRECT SYNTAX: RECT <WIDTH> <HEIGHT>\n");
        return 1;
    }
    width = atof(argv[1]);
    height = atof(argv[2]);
    printf("The rectangle's area is %f\n", width * height);
    printf("The rectangle's perimeter is %f\n", 2 * (width
   + height));
    return 0;
}
```

Bài tập: Đảo ngược câu

- Viết chương trình cho phép người dùng nhập một câu dưới dạng đối số dòng lệnh (mỗi từ trong câu là một đối số). Chương trình hiển thị nội dung câu đảo ngược của câu đã nhập.
- VD: ./inverse I love HUST
- Cho kết quả: HUST love I

Bài tập về nhà

- Viết chương trình tính lũy thừa x của cơ số tự nhiên e^x. Người dùng có thể khởi chạy chương trình với hai cú pháp:
  - <Tên_chương_trình> <giá trị lũy thừa> ví dụ: e 50
  - Hoặc
  - <Tên_chương_trình> <giá trị lũy thừa> <sai_số_cho_phép>. Ví dụ: e 50 0.0003

Bài tập về nhà

- Viết chương trình có tên sde nhận đối số dòng lệnh là các hệ số của một phương trình bậc 2 $ax^2 + bx + c = 0$ và giải phương trình, in ra màn hình các nghiệm.
- Cú pháp sử dụng sde a b c
- Ví dụ: ./sde 1 2 1 cho kết quả: x1=x2 =-1

IV. Làm việc với tập tin

- Tập tin (file) cho phép lưu trữ thông tin lâu dài trong bộ nhớ ngoài.
- Chương trình không chỉ nhập dữ liệu từ bàn phím, hiển thị dữ liệu trên màn hình mà có thể nhập dữ liệu từ tập tin, ghi dữ liệu ra tập tin.
- Ngôn ngữ C cho phép giao tiếp với các tập tin thông qua một thực thể đặc biệt gọi là con trỏ (kiểu) file.
  - Các hàm đọc, ghi tập tin sử dụng con trỏ file là đối số
  - Sau mỗi thao tác đọc/ghi vị trí của con trỏ file sẽ thay đổi.
- Khai báo: FILE *fptr;

Các thao tác cơ bản khi làm việc với tập tin

- Mở tập tin
- Đọc dữ liệu từ tập tin vào bộ nhớ chương trình (cụ thể là vào các biến)
- Ghi (xuất) dữ liệu từ bộ nhớ chương trình ra tập tin
- Đóng tập tin

File văn bản và File nhị phân

File văn bản:

- có nội dung là văn bản chứa các ký tự nhìn thấy được.
- có thể được tạo ra bằng cách dùng các phần mềm thông dụng như Notepad, Notepad++, Sublime Text,…
- thuận tiện trong việc sử dụng hàng ngày, nhưng kém bảo mật và cần nhiều bộ nhớ để lưu trữ hơn.

File nhị phân

- Nội dung lưu trữ dưới dạng nhị phân (chuỗi 0 và 1)
- Bảo mật và tiết kiệm không gian lưu trữ hơn.

Mở tập tin

- Sử dụng hàm fopen()
- Nguyên mẫu: 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 tập tin

- filename: đường dẫn và tên tập tin.
  - Nhận giá trị là một giá trị xâu ký tự: **"data.txt"**
  - Có thể chứa đường dẫn đầy đủ tới tập tin: **"/root/hedspi/CProgrammingBasic/Lab1/data.txt"**
  - Có thể dùng một biến mảng ký tự để lưu trữ:

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

- ***Lưu ý:*** *Nếu đường dẫn không được mô tả, tập tin mặc định nằm tại cùng thư mục chứa chương trình.*

Các tham số mode cho tập tin văn bản

mode	**Ý nghĩa**
"r"	Mở một tập tin văn bản đã có chỉ để đọc. Nếu tập tin không tồn tại, fopen() trả về NULL.
"w"	Mở một tập tin văn bản chỉ để ghi. Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động.
"a"	Mở một tập tin văn bản đã có để ghi thêm vào cuối.
"r+"	Mở một tập tin văn bản đã có cho phép cả đọc và ghi. Nếu tập tin không tồn tại, fopen() trả về NULL.
"w+"	Mở file văn bản cho phép cả đọc và ghi.
"a+"	Mở file văn bản cho phép cả đọc và ghi « append »

Các tham số mode cho tập tin nhị phân

mode	**Ý nghĩa**
"rb"	Mở tập tin nhị phân đã có chỉ để đọc.
"wb"	Mở tập tin nhị phân chỉ để ghi.
"ab"	Mở tập tin nhị phân đã có để ghi thêm vào cuối.
"r+b"	Mở tập tin nhị phân đã có cho phép cả đọc và ghi.
"w+b"	Mở tập tin nhị phân cho phép cả đọc và ghi. Nếu file đã tồn tại, nội dung sẽ bị ghi đè. Nếu file không tồn tại, nó sẽ được tạo tự động.
"a+b"	Mở hoặc tạo tập tin nhị phân cho phép cả đọc và ghi vào cuối.

Đóng tập tin

- Hàm fclose được sử dụng để ngắt kết nối giữa một con trỏ tập tin với tập tin mà nó đang tham chiếu tới.

- int fclose(FILE *stream);

- Cần đóng tập tin khi đã hoàn tất các thao tác đọc ghi với tập tin.

Ví dụ: Đóng và mở tập tin

```
#include <stdio.h>
enum {SUCCESS, FAIL};
main()  {
        FILE *fptr;
        char filename[]= "haiku.txt";
        int reval = SUCCESS;
        if ((fptr = fopen(filename, "r")) == NULL){
                printf("Cannot open %s.\n", filename);
                reval = FAIL;
        } else {
                printf("The value of fptr: 0x%p\n", fptr);
                printf("Ready to close the file.");
                fclose(fptr);
        }
        return reval;
}
```

Đọc và ghi với tập tin (I)

- Trong ngôn ngữ C, chương trình có thể thực hiện các thao tác vào/ra (đọc/ghi) theo những cách thức khác nhau:
  - **Đọc hoặc ghi mỗi lần một ký tự**
  - Đọc hoặc ghi mỗi lần một dòng văn bản.
  - Đọc hoặc ghi một khối (block) các ký tự (byte) mỗi lần.

Thao tác với tập tin theo đơn vị ký tự

- **Đọc hoặc ghi mỗi lần một ký tự.**

- Cặp đôi hàm định nghĩa trong thư viện stdio.h: fgetc() and fputc()

- Nguyên mẫu:
  - int fgetc(FILE *stream);
  - int fputc(int c , FILE *stream);

- Ký tự EOF: khi đến cuối tập tin, các hàm trên trả về EOF.

Bài tập: Sao chép nội dung tập tin

- Tạo một file văn bản với tên lab1.txt với nội dung bất kỳ, lưu trong thư mục cùng với chương trình.
- Viết chương trình đọc từ file trên mỗi lần một ký tự, sau đó ghi chúng vào một file mới với tên lab1w.txt

Đáp án

```
#include <stdio.h>
enum {SUCCESS, FAIL};

void CharReadWrite(FILE *fin, FILE *fout)
{
   int c;
   while ((c=fgetc(fin)) != EOF){
   fputc(c, fout);  /* write to a file */
   putchar(c);
   /* display character on the screen */
   }
}
```

Đáp án

```
enum {SUCCESS, FAIL};

main(void) {
    FILE *fptr1, *fptr2;
    char filename1[]= "lab1a.txt";
    char filename2[]= "lab1.txt";
    int reval = SUCCESS;

    if ((fptr1 = fopen(filename1, "w")) == NULL){
        printf("Cannot open %s.\n", filename1);
        reval = FAIL;
    } else if ((fptr2 = fopen(filename2, "r")) == NULL){
    printf("Cannot open %s.\n", filename2);
    reval = FAIL;
    } else {
        CharReadWrite(fptr2, fptr1);
        fclose(fptr1);
        fclose(fptr2);
    }
    return reval;
}
```

Bài tập: Thao tác với tập tin theo từng ký tự

- Viết chương trình đọc nội dung từ một tập tin văn bản, mỗi lần đọc một ký tự.
- Chương trình sẽ chuyển ký tự chữ cái hoa thành ký tự chữ cái thường và ngược lại, sau đó ghi vào một tập tin khác.
- Chú ý với các ký tự khác – chương trình vẫn thực hiện sao chép một cách thông thường sang tập tin mới.

Đáp án

- Chỉ cần cải tiến nội dung hàm CharReadWrite sử dụng một số hàm xử lý ký tự có trong <ctype.h>

```
void CharReadWrite(FILE *fin, FILE *fout)
{
   int c;
   while ((c=fgetc(fin)) != EOF){
   if islower(c) c=toupper(c);
   else if isupper(c) c=tolower(c);
   fputc(c, fout);  /* write to a file */
   putchar(c); /* display character on the screen */
   }
}
```

Bài tập về nhà

- Viết chương trình có tên mycp hoạt động tương tự lệnh cp trong các hệ điều hành UNIX/LINUX. Nó có thể sao chép một tập tin văn bản sang một tập tin mới theo cú pháp:
  - mycp <tập_tin_1> <tập_tin_2>

- Đường dẫn, tên các tập tin được cung cấp dưới dạng đối số dòng lệnh.
- *Chú ý: Chương trình phải kiểm tra cú pháp sử dụng (vd số đối số – thông báo lỗi và hiển thị hướng dẫn khi cần..)*

Bài tập về nhà: Chương trình nối tập tin

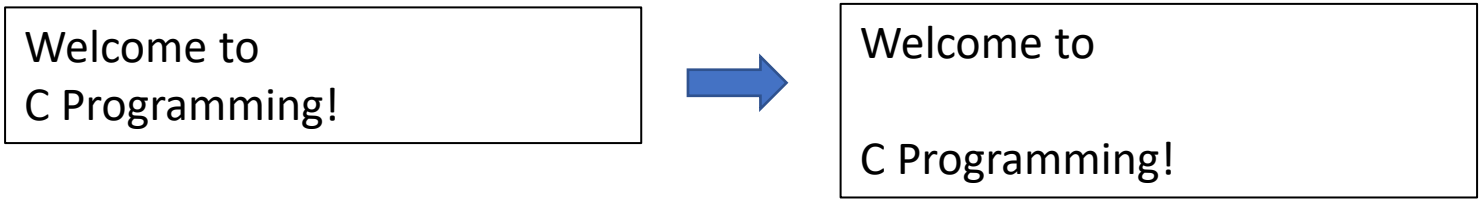
- Viết chương trình nhận tên hai tập tin ở đối số dòng lệnh, sau đó tiến hành ghép nội dung của tập tin thứ hai vào cuối tập tin thứ nhất. Giả sử cả hai tập tin đều tồn tại.
- Cú pháp sử dụng:
  - apd <file1> <file2>
- *Chú ý: Chương trình phải kiểm tra cú pháp sử dụng (vd số đối số – thông báo lỗi và hiển thị hướng dẫn khi cần..)*

Bài tập về nhà: Đổi sang chữ hoa

- Viết chương trình có tên uconvert có chức năng chuyển đổi tất cả các chữ cái trong nội dung một tập tin cụ thể (được cung cấp trong đối số dòng lệnh) thành chữ hoa và ghi lại nội dung mới vào chính tập tin đó.
- Cú pháp: uconvert tata.txt
- Ví dụ
  - File nguồn tata.txt: helloworld
  - Nội dung tata.txt sau khi chạy chương trình : HELLOWORD.

Bài tập

- Viết chương trình double_line nhận một tập tin đầu vào và biến đổi nội dung của nó như sau: chương trình sẽ chèn thêm 1 dòng trống giữa các dòng trong văn bản. Kết quả được ghi vào một tập tin ở đầu ra. Chương trình chạy dưới dạng đối số dòng lệnh như sau: ./double_line <File 1> <File 2>
- Ví dụ minh họa nội dung hai file khi chạy chương trình



Đáp án tham khảo

```
#include <stdio.h>
#include <stdlib.h>

void   double_space(FILE *, FILE *);
void   print_info(char *);

int main(int argc, char **argv) {
   FILE   *ifp, *ofp;

   if (argc != 3) {
      print_info(argv[0]);
      exit(1);
   }
   ifp = fopen(argv[1], "r");      /* mở file 1 để đọc */
   ofp = fopen(argv[2], "w");      /* mở file 2 để ghi */
   double_space(ifp, ofp);
   fclose(ifp);
   fclose(ofp);
   return 0;
}
```

Đáp án tham khảo

```
void double_space(FILE *ifp, FILE *ofp)
{
    int  c;
    while ((c = fgetc(ifp)) != EOF) {
        fputc(c, ofp);
        /* nếu gặp ký tự xuống dòng - nhân đôi nó  */
        if (c == '\n') fputc('\n', ofp);
    }
}
void prn_info(char *pgm_name){
    printf("\n%s%s%s\n\n%s\n\n",
        "Usage:  ", pgm_name, "  infile  outfile",
        "The contents of infile will be double-spaced
    and written to outfile.");
}
```

Bài tập về nhà: Mã hóa Caesar

- Viết một chương trình có thể sử dụng cùng một lúc hai chức năng mã hóa và giải mã một tập tin văn bản sử dụng mật mã Caesar (mã hóa cộng) như sau. Chương trình nhận ba đối số:
  - <tập tin nguồn> <độ dịch chuyển> < tập tin đích>
- Khi cần mã hóa, chạy chương trình với độ dịch chuyển (offset) n là một số nguyên dương. Chương trình sẽ thay thế mỗi ký tự trong tập tin bởi một ký tự đứng sau nó n vị trí trong bảng mã ASCII. Ví dụ với offset =3 thì A→D, B→ E
- Khi giải mã, chạy chương trình với đầu vào là tập tin mã hóa và giá trị độ dịch chuyển là số âm tương ứng (VD offset = -3)
- Chức năng nâng cao (tùy chọn): Với các ký tự là chữ cái thực hiện dịch chuyển vòng tròn: A-> D, ..., Z -> C

Thao tác với tập tin theo dòng

- Sử dụng hai hàm: fgets() and fputs()
- char *fgets(char *s, int n, FILE *stream);
  - s : tham số ứng với xâu ký tự dùng để lưu nội dùng dòng đọc từ tập tin.
  - n : độ dài xâu ký tự s – tính cả ký tự NULL.
- Hàm fgets() dừng khi thỏa mãn một trong các điều kiện sau: đọc được n-1 ký tự từ tập tin, gặp ký tự xuống dòng mới hoặc EOF. Sau đó thêm với ký tự null vào cuối xâu s.

Thao tác với tập tin theo dòng

- Hàm: int fputs(const char *s, FILE *stream);
- s: Xâu ký tự cần ghi ra tập tin
- stream: con trỏ file
- Kết quả trả về
  - 0 nếu thao tác thành công
  - khác 0 nếu thất bại.

Bài tập

- Thực hiện lại bài tập lập trình sao chép nội dung tập tin, tuy nhiên thay vì sử dụng cặp hàm fgetc và fputc – ta sử dụng cặp hàm fgets và fputs để đọc từ tập tin và ghi vào tập tin mỗi lần một dòng trong nội dung văn bản.

Đáp án

```
#include <stdio.h>
enum {SUCCESS, FAIL, MAX_LEN = 81 };

void LineReadWrite(FILE *fin, FILE
  *fout)
{
  char buff[MAX_LEN];
  while (fgets(buff, MAX_LEN, fin) !=
  NULL) {
   fputs(buff, fout);
   printf("%s", buff);
  }
}
```

Đáp án

```
main(void) {
    FILE *fptr1, *fptr2;
    char filename1[]= "lab1a.txt";
    char filename2[]= "lab1.txt";
    int reval = SUCCESS;

    if ((fptr1 = fopen(filename1, "w")) == NULL){
        printf("Cannot open %s.\n", filename1);
        reval = FAIL;
    } else if ((fptr2 = fopen(filename2, "r")) == NULL){
    printf("Cannot open %s.\n", filename2);
    reval = FAIL;
    } else {
        LineReadWrite(fptr2, fptr1);
        fclose(fptr1);
        fclose(fptr2);
    }
    return reval;
}
```

Bài tập: Đếm dòng & hiển thị nội dung (lệnh cat)

- Sửa chương trình sao chép tập tin ở slide trước để chương trình chỉ hiện nội dung tập tin ra màn hình, sau đó hiển thị số các dòng văn bản.
- Minh họa về giao diện của chương trình:

```
Reading file Haiku.txt…. done!
Haiku haiku
Tokyo
Hanoi
This file has 3 lines.
```

Gợi ý

- Sửa mã nguồn hàm LineReadWrite:
  - Loại bỏ lệnh sử dụng fputs
  - Tăng bộ đếm số dòng văn bản mỗi lần đọc một dòng.

```
int LineReadWrite(FILE *fin, FILE *fout)
{
    char buff[MAX_LEN]; int count =0;
    while (fgets(buff, MAX_LEN, fin) != NULL) {
     count++; printf("%s", buff);
    }
    return count;
}
// version 2
int LineReadWrite(FILE *fin)
{
    char buff[MAX_LEN]; int count =0;
    while (fgets(buff, MAX_LEN, fin) != NULL) {
     count++; printf("%s", buff);
    }
    return count;
}
```

Hàm main

```
enum {SUCCESS, FAIL};

main() {
    FILE *fptr1; int c =0;
    char filename1[]= "haiku.txt";
    int reval = SUCCESS;

    if (fptr1 = fopen(filename1, "r")) == NULL){
    printf("Cannot open %s.\n", filename1);
    reval = FAIL;
    } else {
       printf("Reading file %s … done!\n", filename1);
       c= LineReadWrite(fptr2, fptr1);
       printf("The file has %d lines.\n", c);
       fclose(fptr1);
     }
    return reval;
}
```

Bài tập: Hiển thị số thứ tự dòng trong tập tin văn bản

- Viết chương trình đọc một tập tin văn bản và hiển thị ra màn hình số hiệu từng dòng, theo sau là nội dung của dòng đó. Tên tập tin được cung cấp dưới dạng đối số dòng lệnh.
- Ví dụ với tập tin có nội dung

  This is sample file.

  Hello!
- Kết quả trên màn hình là:

  1 This is sample file.

  2 Hello!

Bài tập: Trộn hai tập tin theo dòng

- Viết chương trình có đối số dòng lệnh như sau
  - merge <file1> <file2> <file3>

- Chương trình ghi vào file 3 bằng cách đọc và trộn lần lượt từng dòng từ file 1 và file 2. Chương trình đọc một dòng từ file 1, ghi vào file 3 sau đó đọc một dòng từ file 2 – ghi vào file 3. Chú ý: File 1 và File 2 có thể có số dòng khác nhau, khi đọc hết nội dung một file, chương trình sao chép các dòng tiếp theo của file còn lại vào file 3.

Gợi ý

```
void LineMerge(FILE *f1, FILE *f2, FILE *f3)
{
   char buff1[MAX_LEN], buff2[MAX_LEN];
  int len;
   while ((fgets(buff1, MAX_LEN, f1) != NULL)&&
    (fgets(buff2, MAX_LEN, f2) != NULL)) {
       fputs(buff1,f3);
       fputs(buff2,f3);
   }
   if (buff1!= NULL) fputs(buff1,f3);

   while (fgets(buff1, MAX_LEN, f1) != NULL {
       fputs(buff1,f3);
   }
   while (fgets(buff2, MAX_LEN, f2) != NULL {
       fputs(buff2,f3);
   }
}
```

Bài tập về nhà

- Viết chương trình mycat đọc và hiển thị nội dung một tập tin văn bản trên màn hình. Chương trình hỗ trợ hai cú pháp sử dụng như sau:
- cat <filename> : Hiển thị một lần toàn bộ nội dung
- cat <filename> -p : Hiển thị theo từng trang. Người dùng chạm một phím để xem trang tiếp theo.

Bài tập về nhà

- Viết chương trình nhận đối số dòng lệnh là đường dẫn đến một file văn bản (nội dung dưới 80 dòng). Chương trình thêm một dòng mới vào cuối file nói trên với nội dung chứa các ký tự đầu tiên của các dòng trong file ban đầu.

Đọc và ghi tập tin văn bản với định dạng

- Đây là các hàm hữu ích để xử lý dữ liệu có cấu trúc (thuộc các kiểu dữ liệu khác nhau) từ văn bản.

- int fscanf(FILE *stream, const char *format, ...);
  - Hàm fscanf hoạt động tương tự hàm scanf, điểm khác biệt là nó đọc nội dung từ tập tin (đại diện bởi con trỏ file) để ghi vào các biến.

- int fprintf(FILE *stream, const char *format, ...);
  - Tương tự như hàm printf nhưng thay vì đưa nội dung ra màn hình thì nó ghi nội dung từ các đối số ra tập tin (đại diện bởi con trỏ file).

Bài tập

- Viết chương trình đọc từng dòng văn bản từ một tập tin, sau đó tính độ dài xâu ký tự trên mỗi dòng và ghi ra một tập tin mới theo định dạng sau: <độ dài dòng> <Nội dung dòng>
- Ví dụ, với một dòng trong tập tin đầu vào:

  The quick brown fox jumps over the lazy dog.

  trong tập tin đầu ra ở dòng tương ứng sẽ là:

  44 The quick brown fox jumps over the lazy dog.

Đáp án

Chỉ cần thay đổi hàm LineReadWrite sử dụng hàm strlen và fprintf

```
void LineReadWrite(FILE *fin, FILE *fout)
{
   char buff[MAX_LEN];
   int len;
   while (fgets(buff, MAX_LEN, fin) != NULL)
   {
      len = strlen(buff)-1;
      fprintf(fout, "%d %s", len, buff);
      printf("%s", buff);
   }
}
```

Bài tập

- Viết chương trình để đọc một dãy số nhập từ bàn phím và ghi chúng ra tệp “out.txt” theo thứ tự ngược lại. Ngoài ra, tổng các số được ghi vào cuối file.
- Cú pháp nhập liệu từ bàn phím như sau: Số đầu tiên là số lượng các số trong dãy sẽ nhập, sau đó là dãy các số nguyên. Ví dụ: khi người dùng nhập: 4 12 -45 56 3
- “4” là số các số sẽ được nhập, bao gồm “12 -45 56 3”. Nội dung của tập tin “out.txt” sẽ là, với 26 là tổng của 4 số
- 3 56 -45 12 26
- Vì số lượng các số nhập thay đổi theo mỗi lần chạy, chương trình cần cấp phát động bộ nhớ cho các số này sử dụng hàm malloc( ).

Đáp án

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
//author: Cao Tuan Dung
//for HEDSPI project
enum {SUCCESS, FAIL};

int main(void)
{
    FILE *fp;
    int *p;
    int i,n, value, sum;
    int reval = SUCCESS;

    printf("Enter a list of numbers with the first is
    the size of list: \n");
    scanf("%d", &n);
    p = (int *)malloc(n*sizeof(int)); i=0; sum=0;
```

Đáp án

```
while(i<n) {
        scanf("%d", &value);
        p[i++]=value;
        sum+=value;
    }

    if ((fp = fopen("out.txt","w")) == NULL){
        printf("Can not open %s.\n", "out.txt");
        reval = FAIL;
    }
    for (i=n-1; i>=0;i--){
        fprintf(fp,"%d ",p[i]);
    }
    fprintf(fp,"%d ",sum);
    fclose(fp);
    free(p);
    return reval;
}
```

Bài tập

- Tạo một tập tin văn bản có tên product.txt, mỗi dòng trong đó chứa thông tin về một sản phẩm: ID (kiểu int), Product Name (xâu ký tự không chứa ký tự trắng), Price (kiểu double). Các trường dữ liệu trên phân tách với nhau bởi một ký tự space hoặc tab. Ví dụ
  - 1 Samsung_Television_4K 20000000
  - 2 Apple_MacBook_2020 18560000
- Viết chương trình đọc tập tin trên vào một mảng các phần tử cấu trúc và sau đó hiện nội dung mảng trên ra màn hình dưới dạng:
  - No Product Name Price
  - 1 Samsung_Television_4K 20000000
  - …

Hướng dẫn

- Khi đọc số thực double dùng fscanf
  - "%lf"
- Trong trường hợp các trường dữ liệu được phân tách bởi các ký hiệu như ; hay , (delimiter), có thể kết hợp sử dụng fscanf và fgetc để đọc được từng trường
  - 1000, John_Allan, 28, NewYork

```
#include <stdio.h>
enum {SUCCESS, FAIL, MAX_ELEMENT =
      10};
typedef struct {
     int no;
     char name[20];
     double price;
}product ;
int main(void){
  FILE *fp;
  product arr[MAX_ELEMENT];
  int i=0,n;
  int reval = SUCCESS;
  printf("Loading file...\n");
   if ((fp = fopen("product.txt","r")) == NULL){
       printf("Can not open %s.\n",
       "product.txt");
       reval = FAIL;
  }
   else {
       while (fscanf(fp,"%d%s%lf", &arr[i].no,
arr[i].name, &arr[i].price) != EOF){
        //printf("%-6d%-24s%-
6.2f\n",arr[i].no,arr[i].name,arr[i].price);
                  i++;
       }
       n=i;
       for (i=0; i<n; i++)  printf("%-6d%-24s%-
       6.2f\n",arr[i].no,arr[i].name,arr[i].price);
   }
  fclose(fp);
  return reval;
}
```

Bài tập về nhà

- Tạo một file văn bản nội dung là danh sách lớp gồm ít nhất 6 sinh viên. Mỗi dòng gồm 4 trường sau:
- STT(số thứ tự) Mã số sinh viên Họ và tên (không chứa ký tự trắng) Số điện thoại. Ví dụ
  - 1 20181110 Bui_Van 0903112234
  - 2 20182111 Joshua_Kim 0912123232
- Viết chương trình đọc tập tin trên vào một mảng các cấu trúc phù hợp. Chương trình yêu cầu nhập bổ sung thêm trường điểm cho mỗi sinh viên sau đó ghi lại kết quả vào tập tin bangdiem.txt (transcript.txt) gồm tất cả các trường nói trên (cùng trường điểm).

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict