C Programming Language

JANUARY 2, 2015

Today’s task

- C FILE I/O
- Command line arguments
- Make a simple encryption/decryption program

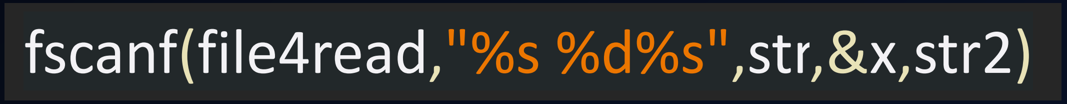
File I/O ---Open a file

```
FILE *file4read       = fopen("test.txt","r");
FILE *file4write      = fopen("out.txt","w");
```

r - open for reading
w - open for writing (file need not exist)
a - open for appending (file need not exist)
r+ - open for reading and writing, start at beginning
w+ - open for reading and writing (overwrite file)
a+ - open for reading and writing (append if file exists)

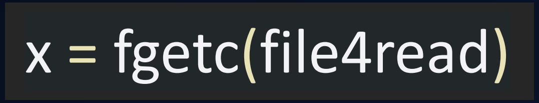
File I/O ---Read a file

- fscanf()
  - read a string from file
  - similar to scanf()



```
fscanf(file4read,"%s %d%s",str,&x,str2)
```

- fgetc()
  - read a character one by one in the file
  - return an int in range of 0~255
  - when at the end of the file, return EOF



```
x = fgetc(file4read)
```

File I/O ---Write a file

- fprintf
    - write a string to the file
    - similar to printf
- fputc
    - write a character to the file
    - x1 should be in range 0~255

```
fputc(x1,file4write);
```

Binary File I/O

```
size_t fread(void *ptr, size_t size_of_elements, size_t
number_of_elements, FILE *a_file);

size_t fwrite(const void *ptr, size_t size_of_elements, size_t
number_of_elements, FILE *a_file);
```

Task

- Open a file (plaintext), read content one by one
- Encrypt the content
- Write in another file (ciphertext)

How to encrypt

- Shift letters
- e.g.
  - key: T(19)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

  - HELLO----------→AXEEH

Plaintext	Ciphertext
H(7)	A((7+19)-26=0)
E(4)	X(4+19=23)
L(11)	E(11+19-26=4)
L(11)	E(11+19-26=4)
O(14)	H(14+19-26=7)

Command Line Argument

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

- argc: the argument count
- argv: a list of the argument variables

Specify the plaintext file and output file

```
int main(int argc, char* argv[])
{
        if(argc != 3)
        {
                printf("usage: %s plaintext outfile",argv[0]);
        }
        else
        {
                FILE *plaintext     = fopen(argv[1],"r");
                FILE *ciphertext    = fopen(argv[2],"w");
                …
        }
        return 0;
}
```

Homework

- ★ We’ ve already make a program to encrypt message, make another program to decrypt the code text to plaintext , also the program should support command line argument
- ★★ The encryption method we just used is called Caesar cipher(http://en.wikipedia.org/wiki/Caesar_cipher), it is old and easy to hack. Think about it, how to hack Caesar cipher?
- ★★★ Caesar cipher is not safe, but another cipher called Vigenère cipher(http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher) based on Caesar cipher is more complex and hard to hack. If you are interested, program it.

Next time

- We make the encryption program in GUI with gtk