

FILE

I-លក្ខណៈទូទៅ :

- ការប្រើប្រាស់ Array រឺ Pointer ទិន្នន័យត្រូវបានដាក់ក្នុងកំណត់ចំណាំ Program រឺពេលដាច់ភ្លើង ។
- ចំពោះទិន្នន័យដែលរក្សាក្នុង File មិនបានដាក់ក្នុងកំណត់ចំណាំ Program រឺពេលដាច់ភ្លើង ។ ហើយទិន្នន័យ

ទាំងនេះ គេអាចទាញយកប្រើការបាននៅពេលក្រោយៗទៀត ។

- File ជាប្រភេទ I/O ដែលកំណត់ដោយ OS ដែលរៀបចំតាម Format ។
- គេរៀបចំ File ដើម្បីអោយ User អាចផ្ទុកទិន្នន័យក្នុង Storage device ដូចជា Hardist រឺ Floppydist ជាដើម ។
- នៅក្នុង C Programming Language File មាន២ប្រភេទគឺ : Binary file និង Text file ។
- File pointer variable ជាប្រភេទនៃ File ដែលកំណត់ក្នុង < stdio.h >

ទំរង់ទូទៅ : FILE *fp ;

II-ការបើក FILE :

ដើម្បីបើក File ប្រើ Function fopen() :

File *fopen(char *filename, char *mode) ;

Ex : fp = fopen(filename , mode) ;

- filename : ជាឈ្មោះ File (ដែលត្រូវបើក) ជា String
- Mode : កំណត់ពីលក្ខណៈរបស់ File អាន សរសេរ Update Data(append) ដែលមាន :

" r " : Open for reading (អាន Data ចេញពី File)

" w " : Open for writting (សរសេរ Data ចូល File)

បើ File មាន Data នោះបណ្តាលអោយ Data ចាស់បាត់បង់ វាបង្កើត File ថ្មីមួយ ។

" a " : Open for appending (បញ្ចូល Data គឺ Data ដែលមានស្រាប់ក្នុង File)

" r+ " : Open file for both reading & writting (អានរឺ Update file ដែលមានស្រាប់)

" w+ " : Open file for both writting & appending (សរសេរ រឺ Append លើ File)

" a+ " : Open file for appending (បញ្ចូល Data បន្តពីមុនសំរាប់ Binary file)

" rb " : Open binary file for reading

" wb " : Open binary file for writting

" ab " : Open binary file for appending

" r + b " : Update binary file for reading

" w+b " : Update binary file for writting

" a+b " : Appending binary file for writing .

សំគាល់ :

- Binary file : សំគាល់តាមរយៈ append ដែលនៅពីក្រោយ Mode របស់វា (" rb " , " r+b ")
- Text file : សំគាល់តាមរយៈ append ពីក្រោយ Mode (" rt " , " r+t " , ...)

១- បើក File សំរាប់បញ្ចូលទិន្នន័យ :

```
FILE *fp ;
fp = fopen("text.dat " , " w " ) ;
if ( fp == NULL ) {
    printf ( "\n could not open file :") ;
    exit (0);
}
```

២- បើក File សំរាប់អានទិន្នន័យ :

```
FILE *fp ;
fp = fopen("text.dat", "r " );
if(!fp){
    print f("could not open file : \n") ;
    exit (0) ;
}
```

៣- បើក File សំរាប់ Append Data :

```
File *fp ;
if (fp = fopen("text.dat", "a" ) == NULL ) {
    print f(could not open file \ n") ;
    exit (0) ;
}
```

4- ការបិទ File (Close file) :

ពេលនៅចុងបញ្ចប់របស់ File (EOF) នោះត្រូវបិទ File ។

```
int fclose (FILE *stream) ;
```

Ex : fclose(fp) ;

III-ការAccessលើ File :

- គេប្រើ Function putc() ដើម្បីបញ្ចូល Character ទៅក្នុង file (file pointer រឺ stdout)
- គេប្រើ Function getc() ដើម្បីអាន Character ពី file(file pointerរឺ stdin)

Ex 1 :

អាន Character ពី File(stdin)បន្ទាប់មកបញ្ចូល Characterទាំងនោះទៅ file (stdout) (output លើ Sreen) :

```
#include<stdio.h>
void main( ) {
    char ch ;
    printf("input a character:") ; ch = getc(stdin);
    while( ch!= ' $ ' ) {
        putc(ch , stdout ) ;
        ch = getc(stdin) ;
    }
    getch( ) ;
}
```

វិភាគសរសេរជា Statement :

```
while (ch = getc(stdin) != '$ ' )  
    putc(ch, stdout) ;
```

Ex 2 : បញ្ចូល Characterតាមរយៈ stdin រួចរក្សាទុកចូល File :

```
void main ( ) {  
    char fname[20], ch ; FILE *fp ;  
    printf("input file name: "); gets(fname);  
    if(fp = fopen(fname,"w ") == NULL){  
        printf("could not open: \n") ;  
        exit (0) ;  
    }  
    printf("input character :\n ");  
    while(ch == getc(stdin) != '$ ' )  
        putc(ch , fp) ;  
    fclose(fp) ;  
    getch ( ) ;  
}
```

Ex 3 :

```
void main ( ) {  
    FILE *fp ; char ch; fname[20];  
    print f("input file name : ") ; gets(fname);  
    if (fp = fopen(fname, "r ") == NULL){  
        print f("could not open\n");  
        exit(0);  
    }  
    ch = getc(fp);  
    while(ch!= EOF){  
        put c(ch,std out );  
        ch = getc(fp);  
    }  
    fclose(fp) ;  
    getch( ) ;  
}
```

Funtion សំខាន់ៗមួយចំនួនដែលអនុវត្តជាមួយ File :

- **fopen()** : Open a file
- **fclose()** : Close a file
- **putc()** : Write a character to file
- **getc()** : Read a character to file
- **fseek()** : Seek to a specified byte in file
- **fprint()** : Prints to a file
- **fscanf()** : Reads from a file
- **feof()** : Return true if EOF is reached
- **ferror()** : Return true if a file error occurred
- **rewind()** : Reset the file cursor to beginning of file
- **remove()** : Deletes a file

- **fread()** : Reads blocks of data from a file
- **fwrite()** : Writes blocks of data to a file

standard Streams :

- **stdin** : Standard input (keyboard)
- **stdout** : Standard output (screen)
- **stderr** : Standard error

Ex4 :

Program input Data ក្នុង File មួយបន្ទាប់មកចំណង Data ទាំងនោះទៅក្នុង File មួយទៀត ដោយប្រើ Function

fgetc() និង fputc() :

```
#include<stdio.h>
void main ( ){
    FILE *fp1 , *fp2 ;
    char fname1[20], fname2[20] , ch ;
    printf("input file name : \n") ; gets(fname1);
    fp1 = fopen(fname1,"w ") ;
    if (!fp1) {
        puts("could not open : ") ;
        exit (1);
    }
    while(ch=fgetc(std in ) != EOF )
        fputc(ch, fp1);
    fclose(fp1);
    fp1 = fopen(fname1, "r ") ;
    printf ("input file name2:\n") ; gets(fname2);
    fp2 = fopen(fname2, "w ");
    if (!fp2){
        puts("could not open\n") ;
        exit(1) ;
    }
    while( ch = fgetc(fp1)!= EOF )
        fputc(ch , fp2) ;
    fclose(fp1) ;
    fclose(fp2);
    getch() ;
}
```

ប្រើ Function fscanf(fp, format, varlist) ;

fprintf(fp, format, varlist) ;

Ex 5 :

```
#include<stdio.h>
void main( ) {
    FILE *fp ; char fname[30] ;
```

```

int d ; float f ; char str[30] ;
printf("input file name : ") ; gets(fname) ;
if( !fp = fopen(fname , "w + b ")) {
    printf("could not open : \n" ) ;
    exit ( 1 ) ;
}
while (fscanf(stdin, "%d % f % s ", &d,&f,str)&& d!= -1){
    fscanf(fp, "%d %f %s\n",d, f, str );
    printf(fp,"%d %f %s \n", d , f , str) ;
}

fclose(fp);
printf("output data :\n ");
printf("input file name : "); gets(fname);
if( !fp = fopen(fname , "r + b")){
    print f("could not open \n ");
    exit(1) ;
}

while( fscanf(fp, "%d %f %s ", &d, &f, &s)!= EOF){
    fprintf(stdout , "%d",d);
    fprintf(stdout , "%f", f);
    fprintf(stdout, "%s", str);
}
fclose(fp) ;
getch() ;
}

```

វិធានសរសេរ :

```

while( fscanf(fp, "%d %f %s ", &d,&f,str) != EOF ){
    printf("d = %d\n", d );
    printf("f = %f \n ", f);
    printf("str = %s\n", str);
}

```

ការប្រើ Function fread() និង fwrite() សំរាប់បញ្ចូល និង អានទិន្នន័យពី File

- fread(void *st , sizeof(datatype) , n , FILE *fp) ;
- fwrite(void *st , sizeof(datatype) , n , FILE *fp) ;

Ex : គេមាន Structure :

```

struct data {
    int d ;
    float f ;
    char name[20];
};

```

បើគេមាន Array នៃ Structure ខាងលើ :

Data a[3] = { {15 , 10.5 , "darith"}, {18,21.4,"sok"}, {27,32.6 ,"bunna"} } ;

```
7 fwrite( &a , sizeof(a) ,1 , fp) ;
```

```
for(int i= 0 ; i < 3 ; i ++)  
    fwrite( &a[i], sizeof(data) , 1 , fp ) ;
```

```

5 fread( &b, sizeof(a) , 1 , fp) ;

```

```
while( ! feof(fp)){
    printf( "d = %d\n", k.d);
    print f("f = %f\n", k.f);
    print f("name = %s\n", k.name);
    fread( &k , sizeof(data) , 1 , fp );
}
```

```
FILE *getFile(char *mode){
    char fname[25] ; file *fp ;
    printf("input file name : ") ;
    gets(fname) ;
    if( !fp = fopen(fname , mode))){
        printf("could not open\n");
        exit(0) ;
    }
    return fp ;
}
```

```
void inputData ( ) {
    FILE *f; float m ; data h ;
    f = getFile("wb") ;
    print f("input d!=2002: ");
    scan f("%d", &h.d ) ;
    while( h.d != 2002){
        printf("input f : "); scanf("%f", &m );
        h.f=m ;
    }
}
```

```

        fflush(stdin) ;
        printf("input name : ") ;  gets(h .name);
        fwrite( &h , sizeof(data) , 1 , f );
        printf("input d: (!2002) : ");
        scanf("%d ", &h.d );
    }
    fclose(f) ;
}

```

-Function ដើម្បីអានទិន្នន័យពី File :

```

void  outputData ( ) {
    data  k ; FILE *f ;
    f = getFile("rb");
    printf("display data\n");
    fread( &k , sizeof (data) , 1 , fp );
    while( !feof(fp) ){
        printf("d = %d\n", k.d );
        printf("f = %f\n", k.f );
        printf("name = %s\n", k.name );
        fread( &k , sizeof(data) , 1 , fp );
    }
    fclose(fp) ;
}

```

* **Rewind** : Function ដើម្បីកំណត់ទីតាំងមកផ្នែកខាងដើមនៃ File pointer

ទំរង់ទូទៅ :

```
void  rewind(FILE *fp );
```

* **ftell** : ជា Function សំរាប់ផ្តល់នូវទីតាំង (Position of file) រឺ គេអាចប្រើ ftell សំរាប់ផ្តល់នូវទំហំរបស់ File ។

ហើយរង្វាស់នៃទីតាំងរបស់ File គិតជាចំនួន Bytes បើទីតាំងនៅការចាប់ផ្តើមនៃ File មានទំហំសូន្យ ហើយបើទីតាំងនៅ Bytes ទី២នៃ File នោះមានទំហំ១ ។ ទំរង់ទូទៅ :

```
long int  ftell(FILE *stream) ;
```

Ex : long int l ;
 l = ftell(fp);
 printf(" l = %ld\n", l);

គេអាចប្រើ Function ftell សំរាប់គណនាចំនួន Structures ដែលមានក្នុង File ។

Ex: numberchar = ftell(fp) ;

```
number  structure = numberchar/sizeof(structure_type);
```

បើការប្រើ ftell មាន Error នោះវា return -1 នេះបណ្តាលពីការប្រើ ftell ជាមួយ device ដែលមិនអាច Store data (ដូចជា Keyboard) រឺទីតាំងគឺជាទំហំនៃ long int ។

* **fseek** : ជា Function សំរាប់កំណត់ទីតាំងជា byte នៅក្នុង File មួយ ។

ទំរង់ទូទៅ :

int fseek(FILE *stream, long offset, int wherefrom);

- **offset** ជាទំហំរបស់ file (ជា byte) ដែលត្រូវ Access ទៅកាន់ទីតាំងណាមួយពិតប្រាកដ ។
- **Wherefrom** : ដើម្បីកំណត់នៅខាងដើមទី រឺ នៅទីតាំង current position រឺនៅចុងក្រោយរបស់ File ។

ដែល wherefrom មាន ៣ គឺ :

SEEK_SET រឺ 0 សំរាប់ទីតាំង Beginning

SEEK_CUR រឺ 1 សំរាប់ទីតាំង Current

SEEK_END រឺ 2 សំរាប់ទីតាំង End

Ex: បើយើងចង់កំណត់ File មួយក្នុងទីតាំង 100(100byte) នោះត្រូវសរសេរ : `fseek(fp ,991 , SEEK_SET);`

-បើ Wherefrom ជា SEEK_CUR រឺ 1 :

- + បើ offset អវិជ្ជមាននោះចល័តទីតាំងត្រលប់ក្រោយ (ទៅ Beginning of file)
- + បើ offset វិជ្ជមាន នោះចល័តទីតាំងទៅខាងមុខ (ទៅ End of file)

-បើ Wherefrom ជា SEEK_END រឺ 2 :

- +បើ offset អវិជ្ជមាន នោះចល័តទីតាំងត្រលប់ក្រោយ (ទៅ Beginning of file)

Ex:

```
fseek(fp , 0L , SEEK_SET) ;  
fseek(fp , 0L , SEEK_CUR) ;  
fseek(fp , 0L , SEEK-END) ;
```

-គណនាទំហំរបស់ file (file size) :

```
fseek(fp , 0l , SEEK_END) ;  
printf("file size = %ld", ftell(fp));
```

-បង្ហាញនូវ Record ទី n :

```
fseek(fp , (n-1)*sizeof(data) , SEEK_SET);  
fread( &k , sizeof(data), 1 , fp);  
printf("d = %d\n " , k.a );  
printf("f = %f\n", k.f );
```

-គណនាចំនួន Record របស់ File :

```
fseek(fp , 0l , SEEK_END) ;  
k = ftell(fp) ;  
n = k/sizeof(data) ;
```


វិធាន data ទាំងអស់ពី File :

```
fseek(fp , 0l , SEEK_SET );  
for( i = 0 ; i < n ; i++)  
fread( &a[i], sizeof (data) ,1 , fp) ;
```

***Remove :** ជា Function ដើម្បីលុប File ណាមួយ :

ទំរង់ទូទៅ : **int remove(char *name) ;**

- Return 0 បើការលុបបានសំរេច
- Return ខុសពី 0 បើមាន Error(file រកមិនឃើញ)

Ex: លុប File មួយដែលមានឈ្មោះ file1.dat នោះ :

```
if( remove("file1.dat")  
    printf( "Error , file can't be deleted : " ) ;
```

***Rename :** ជា Function សំរាប់ផ្លាស់ប្តូរឈ្មោះ File មួយ :

ទំរង់ទូទៅ : **int rename(const char *oldname, const char *newname);**

- Return 0 បើការផ្លាស់ប្តូរឈ្មោះបានសំរេច
- Return ខុសពី 0 បើការផ្លាស់ប្តូរឈ្មោះមាន Error

Ex: ប្តូរឈ្មោះ File file1.dat (ជា oldfile) ទៅជា File ថ្មី File2.dat (ជា newfile) នោះ :

```
if(rename("file1.dat") , "file2.dat"))  
    printf("Error , the file can't be renamed : " );
```