1 *main* функцрүү параметр дамжуулах

main функц нь хоёр параметр авч болно. Тэдгээр параметрт програмд ямар нэгэн утга дамжуулагдсан бол хадгалагддаг.

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

Энд:

int argc дамжуулсан аргументуудын тоо.

char *argv[] дамжуулсан аргумент бүрийн утга, хаяган хүснэгт байдлаар өгөгдсөн. Тэгэхээр хүснэгтийн элемент бүр өөр нэг хаягийг зааж болно гэсэн үг.

Хэрэв *a.out* програмыг доорх байдлаар ажиллуулбал.

```
./a.out 100 22 32 mongol
```

argc-ийн утга 5 харин *argv*-ийн утга:

```
argv[0] => "a.out"
argv[1] => "100"
argv[2] => "22"
argv[3] => "32"
argv[4] => "mongol"
```

Параметрийн утгууд бүгд тэмдэгтэн цуваа учир *atoi(char *)* гэх зэрэг функцүүдийг ашиглан бүхэл тоонд хөрвүүлэн ашиглана.

2 Файл

Си хэл дээр бичсэн програмын стандарт оролтод файл доторх зүйлүүдийг өгвөл

```
./a.out < input
```

Мөн стандарт гаралтыг нь файлруу бичвэл

```
./a.out < input > output
```

аргуудыг хэрэглэж болно.

Мөн түүнээс гадна Си хэлэнд файлаас уншиж, файлруу бичих нь үндсэн гурван алхмаас бүрддэг.

1. Файлыг нээх
2. Файлаас унших, бичих бүх ажлыг хийж дуусгах

3. Файлыг хаах

Файлыг хаалгүй орхих нь тухайн файлыг бусад хэрэглэгч хэрэглэх, мэдээлэл дутуу бичих, санах ой зэрэгт асуудал учруулдаг.

2.1 Файл нээх

Файлыг нээхдээ **stdio.h** санд байрлах *fopen* функцийг хэрэглэнэ.

```
fp = fopen(filename, mode);
```

Энд:

fp нь **FILE*** төрлийн хувьсагч (*FILE* төрлийн бүтцийг заах хаяган хувьсагч)

filename нь онгойлгох файлын нэр.

mode Файлыг хэрхэн онгойлгохыг заах *mode* тэмдэгтэн цуваа. Бид файл унших "r", бичих "w" *mode*-уудыг ихэвчлэн хэрэглэнэ.

fopen() функц нь *FILE** төрлийн хаяг буцаах бөгөөд энэ хаяг нь файлтай харьцахад хэрэглэгдэнэ. Хэрэв файл онгойх боломжгүй (онгойлгох зөвшөөрөлгүй, эсвэл тийм файл байхгүй г.м) байвал *NULL* утга буцаана.

```
FILE *fp;
const char *name = "input";
const char *mode = "r";

fp = fopen(name, mode);
if (fp == NULL) {
  printf("%s file-iig ongoilgoh bolomjgui", name);
  exit(1);
}
```

2.2 Файлаас унших эсвэл файлруу бичих

Файлаа онгойлгосны дараа *fscanf()* функцийг ашиглан уншиж, *fprintf()* функцийг ашиглан бичиж болно. Эдгээр хоёр функц яг л *scanf(), printf()* функцүүдтэй адилхан ажиллах ба зөвхөн уншиж, бичиж байгаа файлыг дүрслэх *FILE** төрлийн аргументыг хамгийн эхэнд нэмж өгөх шаардлагатай.

Жишээ нь: a, b хоёр тоог *output* нэртэй файлруу бичвэл.

```
FILE *fp = fopen("output", "w");
fprintf(fp, "%d%d\n", a, b);
```

fscanf(), fprintf() функцүүд нь хэдэн тоо уншиж бичсэнийг буцаадаг. Тэгэхээр дээрх *fprintf()* 2-ийг буцаана гэсэн үг.

Хэрэв тухайн файлын төгсгөл хүртэл тэмдэгтүүдийг уншихыг хүсвэл файлын төгсгөл хүрсэн эсэхийг шалгахад *feof()* функцийг хэрэглэнэ.

```
FILE *fp = fopen("input", "r");
char s[100];

while (!feof(fp)) {
  fscanf(fp, "%s", s);

  // unshisan temdegtei heregleh
}
```

2.2.1 Бусад функцүүд

1. *fread(), fwrite()* функцүүд нь төрөл харгалзахгүйгээр ямар ч санах ойд байгаа мэдээллийг яг тэр чигээр нь файлруу бичдэг.

   ```
   int fread(void *buf, int size, int count, FILE *fp);
   int fwrite(void *buf, int size, int count, FILE *fp);
   ```

   Эдгээр функц нь уншсан/бичсэн элементийн тоо *count*-ийг буцаана. Энд:

   buf Хаанаа өгөгдлийн хадгалах эсвэл хаанаас өгөгдлийн бичих хаяг.

   size Бичих өгөгдлийн нэг нэгжийн хэмжээ.

   count Тус бүр нь нэгж хэмжээтэй хэдэн ширхэг өгөгдөл унших/бичих.

   fp Файл заагч

   Жишээ: 10 сурагчийн мэдээллийн файлд бичих.

   ```
   struct Student a[10];
   //...
   fwrite(a, sizeof(struct Student), 10, fp);
   ```

2. *fseek(FILE *fp, long offset, int origin);* функц нь уншиж/бичиж байгаа файлын байршил заагчийг *origin* байрлалаас эхлэн *offset* байт зайд байрлуулдаг. Ингэснээр файлын яг тэр байрлалаас эхлэн уншиж бичиж болно.

2.3 Файлыг хаах

Тухайн файлыг хэрэглэж дуусаад заавал хаах шаардлагатай. Файлыг хаахдаа *fclose()* функцэд файлын заагчийг дамжуулна.

```
fclose(fin);
```

2.4 Тусгай файл заагч

Хүссэн файлаа онгойлгохоос гадна Си хэл тусгай файл заагчдыг агуулдаг. Эдгээр нь:

stdin Стандарт оролт

stdout Стандарт гаралт

stderr Стандарт алдаа

Эдгээр файлаас уншиж бичих нь жирийн *scanf(), printf()* функцүүдийг хэрэглэхтэй адил.

```
fprintf(stdout, "Hello World!\n");
```

2.5 Дасгал

2.5.1 Ангид

1. Програмд дамжуулсан N ширхэг тоонуудын нийлбэрийг ол. (*main* функцийн параметрээс утгуудаа авна)

2. *input* файл дахь хоёр бүхэл тоог уншиж, нийлбэрийг *output* файлд бич.

3. *count* файлд нийт хэдэн байт мэдээлэл хадгалагдсаныг харуул. (*feof()* функцийг хэрэглэнэ)

4. *student_write* программ нь n ширхэг сурагчийн мэдээллийг хэрэглэгчээс аван файлд бичнэ. *student_read* функц нь тэр файлд бичигдсэн сурагчдын мэдээллийг уншин дэлгэцэд харуулна. (*fwrite(), fread()* функцүүд хэрэглэгдэнэ)