



RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Computer Architecture (CS-211)

Recitation 3

Siwei Mai

Topics

C Programming

- Read String
- Main Method
- Argument Passing
- File Pointer

* Some materials are collected and compiled from previous year's CS 211 lectures and TAs

Sample C Program

```
#include <stdio.h>
int main(){
    int integer;
    float num;
    char str[100];
    printf("Enter an integer: ");
    scanf("%d", &integer);
    printf("Enter a floating number: ");
    scanf("%f", &num);
    printf("Enter a string: ");
    scanf("%s", str);

    printf("Int : %d\nFloat : %f\nString : %s\n", integer, num, str);
    return 0;
}
```

String as an Input - 1

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

```
int main()
```

```
{
```

```
    char str[20];
```

```
    gets(str); // deprecated (The most unsafe method, buffer overflow will happen)
```

```
    printf("%s", str);
```

```
    return;
```

```
}
```

```
// Segmentation fault for more than 20 chars
```

String as an Input - 2

```
#include <stdio.h>
#define MAX_LIMIT 20
int main()
{
    char str[MAX_LIMIT];
    fgets(str, MAX_LIMIT, stdin);
    printf("%s", str);

    return;
}
```

// No segmentation fault for 19+ chars

String as an Input - 3

```
#include <stdio.h>
int main()
{
    char str[5];
    scanf("%s", str);
    printf("%s", str);

    return;
}
```

// can store more than 5 chars, why?

String as an Input - 4

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

```
int main()
```

```
{
```

```
    char str[5];
```

```
    scanf("%4s", str);
```

```
    printf("%s", str);
```

```
    return;
```

```
}
```

```
// can not store more than 5 chars
```

String as an Input - 5

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

```
int main()
```

```
{
```

```
    char str[5];
```

```
    scanf("%[^\\n]%*c", str);
```

```
    printf("%s", str);
```

```
    return;
```

```
}
```

- %*[^\\n] scans everything until a \\n, but doesn't scan in the \\n. The asterisk(*) tells it to discard whatever was scanned.
- %*c scans a single character, which will be the \\n left over by %*[^\\n] in this case. The asterisk instructs scanf to discard the scanned character.

// can store more than 5 chars

String as an Input - 6

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[]="abcde";
    printf("size:%s\n", sizeof(str));
    printf("len:%s\n", strlen(str));

    return 0;
}
```

// What is the results?

Main Method

```
// test_main1.c
#include <stdio.h>
int main( int argc, char * argv [] ) {

    printf( "argc = %d\n", argc );
    int i;
    for(i = 0; i < argc; i++ ) {
        printf( "argv[ %d ] = %s\n", i, argv[ i ] );
    }
}
```

```
// run and see the
$ gcc test_main1.c -o test_main1
$ ./test_main1 Hi 211 CS
argc = 4
argv[ 0 ] = ./test_main1
argv[ 1 ] = Hi
argv[ 2 ] = 211
argv[ 3 ] = CS
```

```
// same program
#include <stdio.h>
int main( int argc, char **argv) {

    printf( "argc = %d\n", argc );
    int i;
    for(i = 0; i < argc; i++ ) {
        printf( "argv[ %d ] = %s\n", i, argv[ i ] );
    }
}
```

Reading Files

```
#include<stdio.h>
int main()
{
    FILE *ptr_file;
    char buf[1000];

    ptr_file =fopen("input.txt","r");
    if (!ptr_file)
        return 1;

    while (fgets(buf,1000, ptr_file)!=NULL)
        printf("%s",buf);

    fclose(ptr_file);
    return 0;
}
```

*char *fgets(char *str, int n, FILE *stream)*

str – This is the pointer to an array of chars where the string read is stored.

n – This is the maximum number of characters to be read (including the final null-character). Usually, the length of the array passed as str is used.

stream – This is the pointer to a FILE object that identifies the stream where characters are read from.

Q&A

Thank you!