

TI DSP, MCU, Xilinx Zynq FPGA Based Programming Expert Program

Instructor – Innova Lee(Sanghoon Lee)

gcccompil3r@gmail.com

Student – Hyungju Kim

mihaelkel@naver.com



What I learned today(03.19)

System call :

the only interrupt made by software

Philosophy of Linux :

"Everything is a file", more exactly saying, "Everything is a file descriptor".

file descriptor : It has the type of int(integer), starting from 0. 0 to 2 is fixed. 0 means standard in(stdin), 1 means standard out(stdout), 2 means standard error(stderr). So, If you open a file, the file's fd(file descriptor) be allocated from 3.

A few System call :

open, read, write, create, close etc.

open(path, flags, mode)

path : file name(char*)

flags :

O_CREAT : create

O_TRUNC : truncation

O_RDONLY : read only

O_WRONLY : write only

O_RDWR : write and read

mode : 0644

Return type : int (file descriptor)

creat(path, mode)

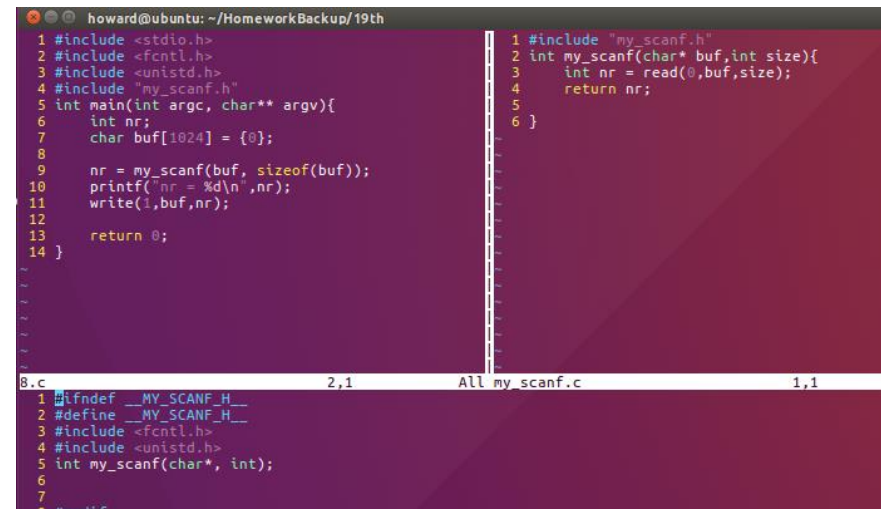
It is the same as open, defaulted O_CREAT, O_TRUNC, O_WRONLY

read(fp, buf, sizeof(buf))

write(fp, buf, length)

Separating a file :

Can you tell me why We use function? The answer is, for modulization. When the code is too long, using function is not enough to managing the program. So, there is a method, separating a file(making header file).



```
howard@ubuntu: ~/HomeworkBackup/19th
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include "my_scanf.h"
5 int main(int argc, char** argv){
6     int nr;
7     char buf[1024] = {0};
8
9     nr = my_scanf(buf, sizeof(buf));
10    printf("nr = %d\n", nr);
11    write(1, buf, nr);
12
13    return 0;
14 }

1 #include "my_scanf.h"
2 int my_scanf(char* buf, int size){
3     int nr = read(0, buf, size);
4     return nr;
5 }
6 }
```

Multi-view instructor :

:vs -> making multi-view horizon direction.

:sp -> making multi-view vertical direction.

:e filename -> open another file.

