KWANGWOON UNIVERSITY

2023년 1학기 **시스템프로그래밍실습** 6주차

Final ls

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Contents

- ls
- Wild card matching
- Lab.

ls

- **Data types**
  - DIR, struct dirent, struct passwd, struct stat, struct tm

- **System Calls & Functions**
  - opendir(), readdir(), closedir()
  - stat()
  - getgrgid(), getpwuid()
  - localtime()
  - getcwd()
  - getopt()
  - **fnmatch()**

Wild card matching

- **fnmatch()**

```
#include <fnmatch.h>

int fnmatch(const char *pattern, const char *string, int flags);
```

  - Description
    - Checks whether the string argument matches the pattern argument, which is a shell wildcard pattern ('*', '?', '[seq]').
  - Return value
    - If match, zero
    - If not match, nonzero(FNM_NOMATCH)
  - Shell wildcard pattern
    - '*': matches everything
    - '?': matches any single character
    - '[seq]': matches any single character in seq

Wild card matching (cont'd)

- **fnmatch()**
  - Example: shell's wildcard pattern

```
~/spls$ ls
ls.c ls.h Makefile spls_final text1.txt text2.txt

~/spls$ ls *
ls.c ls.h Makefile spls_final text1.txt text2.txt

~/spls$ ls *.c
ls.c

~/spls$ ls ????  물음표개수에 맞는 글자수 출력 네글자 출력
ls.c ls.h

~/spls$ ls *.?
ls.c ls.h

~/spls$ ls *.[a-z]  a-z중 한글자
ls.c ls.h

~/spls$ ls text[12].txt    1,2중하나
text1.txt text2.txt

~/spls$ ls *.[^t]  t제외하고 출력
ls.c ls.h
```

Wild card matching (cont'd)

- **fnmatch()**
  - Bitwise OR of zero or more of the following flags
    - 0
      - No flags
      - Passing this value for our assignment
    - FNM_NOESCAPE
      - '₩' 문자를 일반 문자로 취급(e.g., ₩?)
    - FNM_PATHNAME
      - '/' 을 단순 문자가 아닌 패턴으로 인식
    - FNM_PERIOD
      - 문자열의 처음에 '.' 문자가 나타나면 특별하게 취급

Lab.

- **Code**
  - fnmatch.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <fnmatch.h>

int main(int argc, char** argv){
        if(argc < 3){
                printf("argc != 3 \n");
                return 0;
        }
        printf("%s and %s is ", argv[2], argv[1]);
        if (!fnmatch(argv[2], argv[1], 0))
                printf("matching\n");
        else
                printf("not matching\n");

        return 0;
}
```

Lab. (cont'd)

- **Result**

```
sslab@ubuntu:~/final_is$ ls
fnmatch  fnmatch.c
sslab@ubuntu:~/final_is$ ./fnmatch
argc != 3
sslab@ubuntu:~/final_is$ ./fnmatch 2 '?'
? and 2 is matching
sslab@ubuntu:~/final_is$ ./fnmatch 123 '?2?'
?2? and 123 is matching
sslab@ubuntu:~/final_is$ ./fnmatch kkk '*'
* and kkk is matching
sslab@ubuntu:~/final_is$ ./fnmatch a.txt '*.txt'
*.txt and a.txt is matching
sslab@ubuntu:~/final_is$ ./fnmatch text1 'text[1-9]'
text[1-9] and text1 is matching
sslab@ubuntu:~/final_is$ ./fnmatch test10 'text[1-9]'
text[1-9] and test10 is not matching
sslab@ubuntu:~/final_is$ ./fnmatch aabbccdd 'a?b?c?dc'
a?b?c?dc and aabbccdd is not matching
```

Lab. (cont'd)

- **Result 2**

Set flag

```
if (!fnmatch(argv[2], argv[1], FNM_PATHNAME))
        printf("matching\n");
else
        printf("not matching\n");
```

```
① sp2018722000@ubuntu:~/sslab$ ./fnmatch make/dir make*
make* and make/dir is not matching
sp2018722000@ubuntu:~/sslab$ vim fnmatch.c
sp2018722000@ubuntu:~/sslab$ make
gcc -o fnmatch fnmatch.c
② sp2018722000@ubuntu:~/sslab$ ./fnmatch make/dir make*
make* and make/dir is matching
sp2018722000@ubuntu:~/sslab$
```

① Use 'FNM_PATHNAME' flag → Not Matching

② Same as Sample Code 1 → Matching