

2024년 1학기 시스템프로그래밍실습 11주차

FTP3-1

System Software Laboratory
Dept. of Computer Engineering,
Kwangwoon Univ.

과제 세부 일정

과제 차수	세부 차수	강의 및 구현 내용
1	1	String 관련 함수, getopt 함수
	2	ls 명령어 구현을 위한 file system
	3	파일 관련 명령어 구현 함수, Server, client 구현 및 FTP Simulation
2	1	소켓 프로그래밍
	2	fork 함수 및 시그널 관련 함수 (1)
	3	fork 함수 및 시그널 관련 함수 (2)
3	1	User authentication/access control
	2	Split connection/transmission mode
	3	Log file

User Authentication

System Software Laboratory
Dept. of Computer Engineering,
Kwangwoon Univ.

User Authentication (1/6)

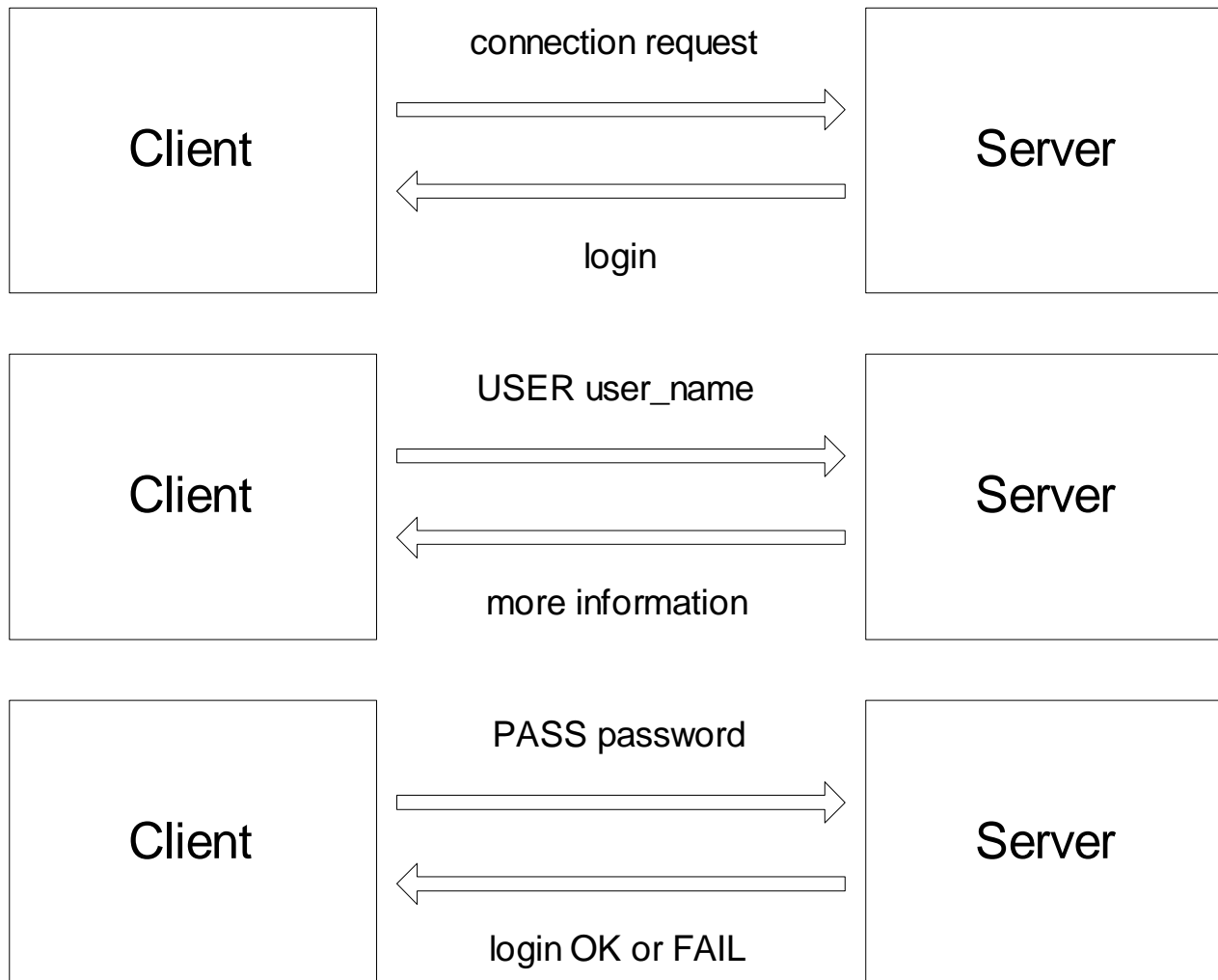
▶ Client

- ▶ Pass user name and password
 - ▶ Send user name and password
 - ▶ User name and password are not encryption

▶ Server

- ▶ Check the user name and password
 - ▶ Receive user name and password
 - ▶ Find name and password in pre-defined text file
 - ▶ If no match then send error message.
 - ▶ If third time no match then session close

User Authentication (2/6)



User Authentication (3/6)

▶ Related Function – Turn off echoing to get password

▶ Synopsis

```
#include <unistd.h>

char *getpass (const char * prompt);
```

▶ Return Value

- ▶ Pointer **; if OK**
- ▶ NULL **; on error**

▶ Example

```
main() {
    char *passwd;
    passwd = getpass("input passwd : ");
    printf("passwd : %s\n", passwd);
}
```



```
$ ./login
input passwd : # 123 입력
passwd : 123
$
```

User Authentication (4/6)

▶ Related Function – Get password file entry

▶ Synopsis

```
#include <stdio.h>
#include <pwd.h>
#include <sys/types.h>

struct passwd *fgetpwent (FILE *stream);
```

▶ Return Value

- ▶ *passwd* structure pointer ; if OK
- ▶ NULL ; on error and EOF

```
struct passwd {
    char    *pw_name;           /* username */
    char    *pw_passwd;        /* user password */
    uid_t   pw_uid;            /* user ID */
    gid_t   pw_gid;            /* group ID */
    char    *pw_gecos;         /* real name */
    char    *pw_dir;           /* home directory */
    char    *pw_shell;         /* shell program */
};
```

User Authentication (5/6)

▶ Related Function – Get password file entry (cont'd)

▶ File structure to be parsed

▶ same as */etc/passwd*

▶ Syntax

□ {user_name}:{password}:{user_ID}:{group_ID}:{real_name}:{home_directory}:{shell_program}

▶ E.g.


```
test1:12:0:0:SPLab1:/home/1:sh1  
test2:34:1:0:SPLab2:/home/2:sh2  
test3:56:2:0:SPLab3:/home/3:sh3
```


User Authentication (6/6)

▶ Related Function – Get password file entry (cont'd)

▶ Example

```
int main(void) {  
    FILE *fp; // 파일구조체에 포인터 변수를 선언  
    struct passwd *pw;  
  
    fp = fopen("passwd", "r"); // passwd 파일을 읽기전용으로 개방  
    while((pw = fgetpwent(fp)) != NULL) {  
        printf("user:%s pw:%s uid:%d gid:%d\n",  
               pw->pw_name, pw->pw_passwd, pw->pw_uid, pw->pw_gid);  
        printf("name:%s home_dir:%s shell:%s\n\n",  
               pw->pw_gecos, pw->pw_dir, pw->pw_shell);  
    }  
  
    fclose(fp);  
    return 0;  
}
```



```
user:test1 pw:12 uid:0 gid:0  
name:SPLab1 home_dir:/home/1 shell:sh1  
  
user:test2 pw:34 uid:1 gid:0  
name:SPLab2 home_dir:/home/2 shell:sh2  
  
user:test3 pw:56 uid:2 gid:0  
name:SPLab3 home_dir:/home/3 shell:sh3
```

Access Control

System Software Laboratory
Dept. of Computer Engineering,
Kwangwoon Univ.

Getting IP Address of Client

▶ *sockaddr_in* structure 내부의 *sin_addr* 이용

```
struct sockaddr_in
{
    short sin_family;
    u_short sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
}
```

▶ IPv4 주소를 Dotted-decimal 문자열로 변환

```
#include <arpa/inet.h>

char *inet_ntoa (struct in_addr inaddr);
```

▶ Returns

- ▶ pointer to dotted-decimal string
- ▶ E.g.
 - "128.134.52.62"

Access Control of Server (1/2)

▶ Conditions

- ▶ 연결을 요청한 client의 IP를 기준으로 판단
- ▶ Access 가능한 client의 list를 file(***access.txt***)로 유지
- ▶ List에 존재하지 않을 경우 error message 전송
 - ▶ 이 때, 해당 IP address를 message에 명시
- ▶ Wildcard(*)를 사용할 수 있도록 해야 함
 - ▶ 단, 숫자 중간에 wildcard가 들어가는 경우는 생각하지 않아도 됨.
 - E.g. 128.134.52.* ➔ **구현해야 함**
 - 128.134.52.*2 ➔ **구현하지 않아도 됨.**

▶ Example of ***access.txt***

```
128.134.64.*
128.134.57.*
128.134.54.171
203.245.*.*
203.238.133.*
```

Access Control of Server (2/2)

► E.g.

