

프로그래밍 연습

2017-09-12

2주차 실습

실습환경

- Martini.snucse.org
 - Terminal 실행
 - ssh [{id}@martini.snucse.org](ssh:{id}@martini.snucse.org)
 - 컴퓨터공학과 전공, 부전공 생들은 학과에서 발급된 자신의 계정을 사용
 - passwd 명령어를 넣어서 비밀번호를 수정하자
- http://mrl.snu.ac.kr/courses/CourseProgrammingPractice/PP_2017fall.htm
에서 자신의 계정을 확인

Shell Command

- **mkdir (디렉토리 이름)** : 새로운 디렉토리 생성
 - Ex) mkdir programming => programming이라는 이름의 디렉토리 생성
- **ls** : 현재 디렉토리에 있는 파일 목록 보기
- **cd (디렉토리 이름)** : 해당 디렉토리로 이동
 - ./ 현재 디렉토리
 - ../ 부모 디렉토리
 - ~/ 홈 디렉토리
- **rm (파일 이름)** : 해당 파일 삭제

Shell Command

- **mv (argument 1) (argument 2) :**
 - argument 1 파일을 argument 2로 옮기거나, argument 1 파일의 이름을 **argument 2로 변경**
 - Ex) mv programming.c programming_dir/
 - programming.c 파일을 programming_dir로 옮김
 - mv programming.c programming2.c
 - programming.c 파일의 이름을 programming2.c로 바꿈
- **cp (argument 1) (argument 2) :**
 - Argument 1 파일을 argument 2 이름을 가진 파일로 복사

Shell Command (추가)

- **./a.out < input**

“input”이라는 파일 안에 써져 있는 정보를 input으로 대입한다

- **./a.out > output**

a.out의 실행 결과가 “output”이라는 파일 안에 써지게 한다. 덮어쓰기임을 주의하세요

- **man {command name}**

특정 command의 사용법(manual) 보기

- **구글에 shell command 검색!**

Vim 편집기

- 초기 리눅스 사용시

```
ubuntu@ubuntu:~$ vim
The program 'vim' can be found in the following packages:
* vim
* vim-gnome
* vim-tiny
* vim-athena (You will have to enable component called 'universe')
* vim-athena-py2 (You will have to enable component called 'universe')
* vim-gnome-py2 (You will have to enable component called 'universe')
* vim-gtk (You will have to enable component called 'universe')
* vim-gtk-py2 (You will have to enable component called 'universe')
* vim-gtk3 (You will have to enable component called 'universe')
* vim-gtk3-py2 (You will have to enable component called 'universe')
* vim-nox (You will have to enable component called 'universe')
* vim-nox-py2 (You will have to enable component called 'universe')
Try: sudo apt install <selected package>
```

- Vim이 설치 되지 않은 것이므로 'sudo apt install vim' 명령어를 통해 설치를 해야함

Vim 편집기

- vim {파일명.확장자}
 - 새로운 파일 만들기 / 기존 파일 열기
 - 프로그램 언어에 따라 확장자 설정(e.g., test.c)
- vim 편집기에서 알아야 하는 mode
 - normal mode
 - vim에 들어가자마자 나오는 첫 번째 모드
 - insert mode
 - normal mode에서 'i'를 누르면 insert mode로 진입
 - 편집모드
 - 'esc'를 누르면 normal mode로 돌아감
- vim 편집기를 켜고 'i'를 눌러서 insert mode에서 편집하기

Vim 편집기

- normal mode에서의 명령어
 - :w 저장하기
 - :q 나가기
 - :wq 저장하고 나가기
 - :q! 바뀐 내용 저장하지 않고 나가기
 - u undo
 - ctrl+r redo
 - d{x}d x줄 잘라내기
 - y{x}y x줄 복사
 - p 복사한 내용 붙여넣기
- <http://www.joinc.co.kr/w/Site/Vim/Documents/UsedVim> 참고!

vi / vim 단축키 모음

Esc

명령 모드

~ 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄 끝으로 이동	% 일치하는 괄호 찾기	^ 줄의 첫 글자	& :8 번복	* 다음 검색	(문장 시작) 문장 끝	_ 아래줄로 이동	+ 다음 줄
· 매크로 이동	1	2	3	4	5	6	7	8	9	0 줄의 처음	- 이전 줄	= 자동 들여쓰기 ³
Q 실행 모드	W 다음 WORD	E 끝 WORD	R 수정 모드	T 위로 검색	Y 줄단위 복사	U 줄 단위 실행취소	I 줄 시작에서 삽입	O 행 위에 삽입	P 커서 이전에 붙여넣기	{ 문단 시작	}	문단 끝
q 매크로 기록	w 다음 단어	e 단어 끝	r 한 문자 교체	t 한 문자 검색	y 복사	u 실행취소	i 편집 모드	o 행 아래에 삽입	p 커서 이후에 붙여넣기	[기타]	기타
A 줄 끝에 덧붙이기	S 줄 삭제후 편집모드	D 줄 끝까지 삭제	F 위로 검색	G 파일끝/줄로 이동	H 화면 상단	J 줄 합치기	K 도움말	L 화면 하단	: ex 명령줄	" 레지스터 지정	열 이동	
a 덧붙이기	s 단어 삭제후 편집모드	d 삭제	f 한 문자 찾기	g 확장 명령	h ←	j ↓	k ↑	l →	; t/T/f/F 명령 반복	' 매크로 이동	\ 사용 안함	
Z 종료 ⁴	X 백스페이스	C 줄 끝까지 바꾸기	V 줄단위 비주얼 모드	B 이전 WORD	N 이전 (찾기)	M 화면 가운데	< ³ 내어쓰기	> ³ 들여쓰기	? 찾기 (뒤로)			
Z 확장 명령 ⁵	X 글자 삭제	c 바꾸기	v 비주얼 모드	b 이전 단어	n 다음 (찾기)	m· 마크 설정	, t/T/f/F 역순 검색	· 명령 반복	/· 찾기			

동작

커서를 이동하거나, 연산자가 동작할 범위를 지정합니다.

명령

바로 동작하는 명령, 빨간색은 편집 모드로 변경됩니다.

연산자

이동 관련 문자(숫자나 커서 이동)와 함께 사용하여 하며, 커서의 위치부터 목적지까지 연산합니다.

확장

특별한 키 합수로, 추가적인 키 입력이 필요합니다.

q·

입력후 (숫자를 제외한)으로 끝낼수 있는 글자를 입력하여야 합니다.

words: 구분자로 공백, 특수기호 모두 사용
WORDS: 구분자로 공백 문자만 사용

words: quux(foo, bar, baz);

WORDS: quux(foo, bar, baz);

주요 명령행 명령 ('ex'):

:w (저장), :q (종료), :q! (저장하지 않고 종료)
:e f (파일 f 열기),
:%s/x/y/g (파일 전체에서 'x' 를 'y' 로 교체),
:h (vim 도움말), :new (새 파일)

그외 중요한 명령들:

CTRL-R: 재실행 (vim),
CTRL-F/-B: 페이지 위로/아래로,
CTRL-E/-Y: 줄 스크롤 위로/아래로,
CTRL-V: 블록비주얼 모드 (vim 전용)

비주얼 모드:

커서를 움직여 지정한 범위에 연산자를 적용합니다. (vim 전용)

참고:

- (1) 복사/붙여넣기/지우기 명령어를 사용하기 전에 "x"를 입력하여 레지스터(클립보드)를 지정하세요. (x는 a에서 z 또는 * 을 사용할 수 있음) (예: "ay\$ 를 입력하면 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.)
- (2) 어떤 명령을 입력하기 전에 횟수를 지정하면, 횟수만큼 반복하게 됩니다.(예: 2p, d2w, 5l, d4j)
- (3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다. 예시: dd(현재 라인 지우기), >>(들여쓰기)
- (4) ZZ 는 저장후 종료, ZQ는 저장하지 않고 종료.
- (5) zt : 커서가 위치한 곳을 제일위로 올리기, zb : 바닥으로, zz : 가운데로
- (6) gg : 파일의 처음으로(Vim 전용), gf : 커서가 위치한 곳의 파일 열기(Vim 전용)

Shell Command Tutorial

- "PP2016", "0921"라는 폴더를 만들어보자
 - > mkdir PP2016
 - > mkdir 0921
- "0921" 폴더를 "PP2016"폴더 하위에 이동시키자
 - > mv 0921 ./PP2016/0921
- "PP2016/0922" 폴더에 진입하자
 - > cd PP2016/0921
- "vim test.c"를 통해 test.c 코드를 작성했다고 가정하자
 - > vim test.c
- "cat test.c"를 통해 test.c 코드를 command 창을 통해서 보자
 - > cat test.c

Shell Command Tutorial

- "ls"를 통해 폴더 내 요소를 살펴보자

```
> ls
```

- "rm test.c" 를 통해 지워보자

```
> rm test.c
```

Lab Assignment

1

숫자를 입력이 끝날 때까지(EOF까지) 입력을 받아서 다음에서 요구하는 것을 출력하라. 단, 숫자는 [0, 2147483647(int 최대값)] 사이임이 보장된다.

- a) 입력된 숫자 중에서 2번째로 큰 숫자를 출력하라.(중복 제거)
만약 없다면 -1 을 출력하라.
- b) 입력된 숫자 중에서 3번째로 큰 숫자를 출력하라.(중복 제거)
만약 없다면 -1 을 출력하라.
- c) 입력된 숫자들의 평균을 소수형태(%f)로 출력하라.

1-a 예제

입력 : 1 2 3 4 5

출력 : 4

입력 : 1 2 1 2 1 2

출력 : 1

입력 : 123 123 123

출력 : -1

1-b 예제

입력 : 1 2 3 4 5

출력 : 3

입력 : 1 2 3 1 2 3 1 2 3

출력 : 1

입력 : 123 123 123 456 456

출력 : -1

1-c 예제

입력 : 1 2 3 4 5

출력 : 3.000000

입력 : 0 1 1

출력 : 0.666667

입력 : 2147483647 2147483646 2147483645

출력 : 2147483646.000000

최대값을 구하는 코드

- 아래는 수업시간에 진행한 최댓값을 구하는 코드이다.

```
#include <stdio.h>
int main(){
    int x, max;
    max=-1;
    while (scanf("%d",&x)!=EOF){
        if (x>max){
            max=x;
        }
    }
    printf("%d\n",max);
    return 0;
}
```

과제 실행 방법

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

Or

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

과제

- 이번 과제는 수업시간 내에 끝내지 못하시면 아래 메일로 제출하셔야 합니다.

snupp2017@gmail.com

※ 제출 서식 (이후의 모든 과제도 동일합니다.)

수업시간에 완료하지 못한 코드를 파일 별로 제출해주세요.

파일명) 20XX-OOOOO_02_0x.c

ex) 2014-16371_02_01.c (02번째 실습의 01번째 과제)

메일명) [SNUPP01-01]20XX-OOOOO_이름_xx

ex) [SNUPP01-01]2014-16371_류호석_02 (마지막 02는 02번째 실습을 의미합니다. 하이픈, 언더바 구분해주세요)

+ 추가 vimrc

vim 편집기 설정을 바꾸는 파일

vim ~/.vimrc

자주 쓰는 설정

- set number "line 표시

- set autoindent "자동 인덴트

- set smartindent "스마트 인덴트

- set cindent "c style의 인덴트

- set shiftwidth=4 "shift를 4칸으로

- set tabstop=4 "tab을 4칸으로

- set ruler "상태표시줄에 커서의 위치 표시

구글에 vimrc를 검색하면 더 많은 설정을 할 수 있어요!

+추가 putty

실습실은 linux os라서 바로 터미널에서 서버에 접속할 수 있지만 windows에서 과제를 하려면 리눅스 서버 원격 접속 프로그램이 필요하다.

구글에 putty 검색, 다운로드

+추가 putty

Host Name

martini.snucse.org
{id}@martini.snucse.org

Port

22

Connection type

SSH

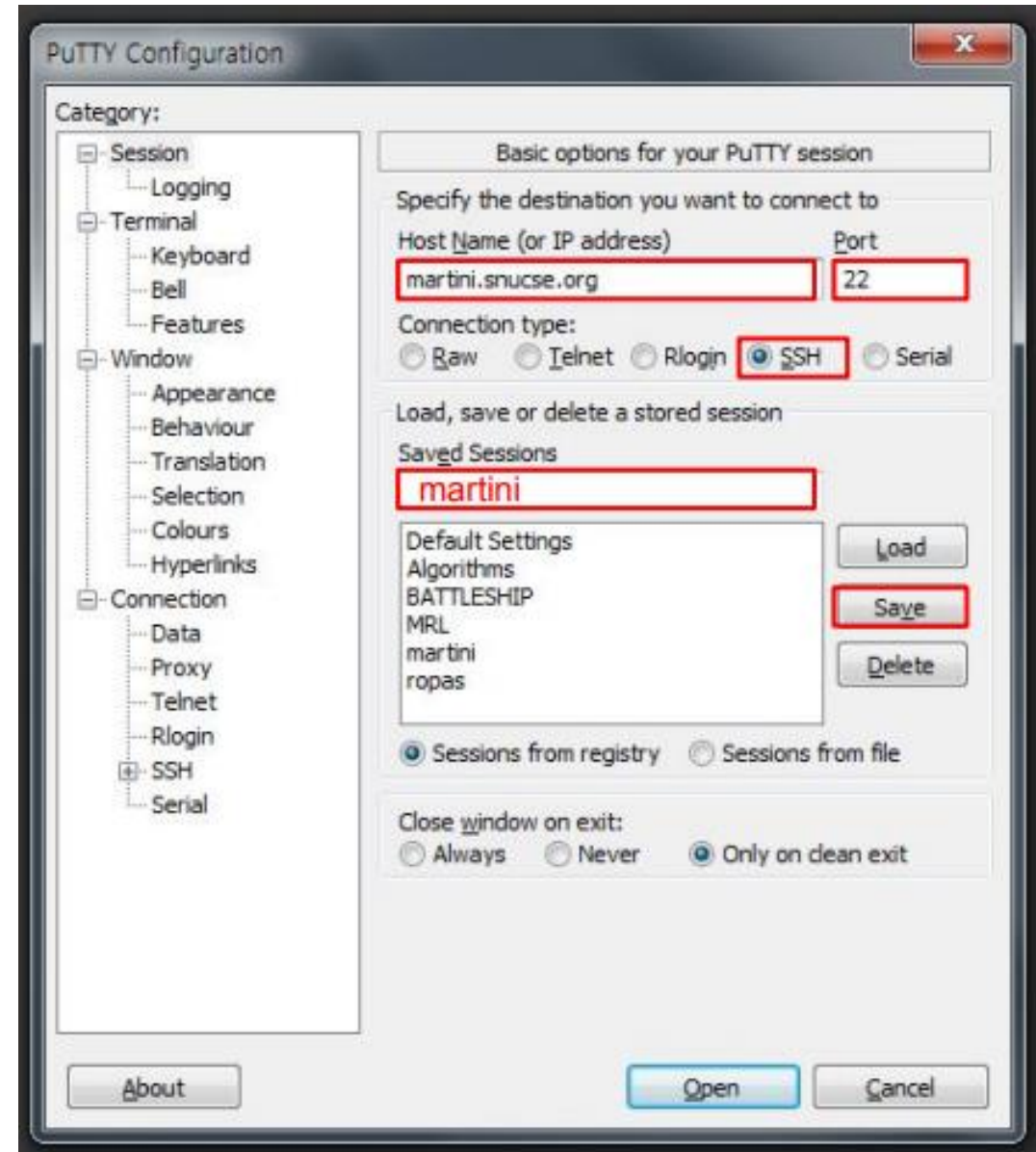
Saved Session

session명 마음대로 설정

Save

session 저장

session명 더블 클릭으로 접속



Google

Filezilla



FileZilla

The free FTP solution

[Home](#)

FileZilla

[Features](#)
[Screenshots](#)
[Download](#)
[Documentation](#)

FileZilla Server

[Download](#)

Community

[Forum](#)
[Project page](#)
[Wiki](#)

General

[Contact](#)
[License](#)
[Privacy Policy](#)

Development

Overview

Welcome to the homepage of FileZilla, the free FTP solution. Both a client and a server are available. FileZilla is open source software distributed free of charge under the terms of the GNU General Public License

Support is available through our [forums](#), the [wiki](#) and the [bug and feature request trackers](#).

In addition, you will find documentation on how to compile FileZilla and nightly builds for multiple platforms in the development section.

Quick download links

**Download
FileZilla Client**

All platforms

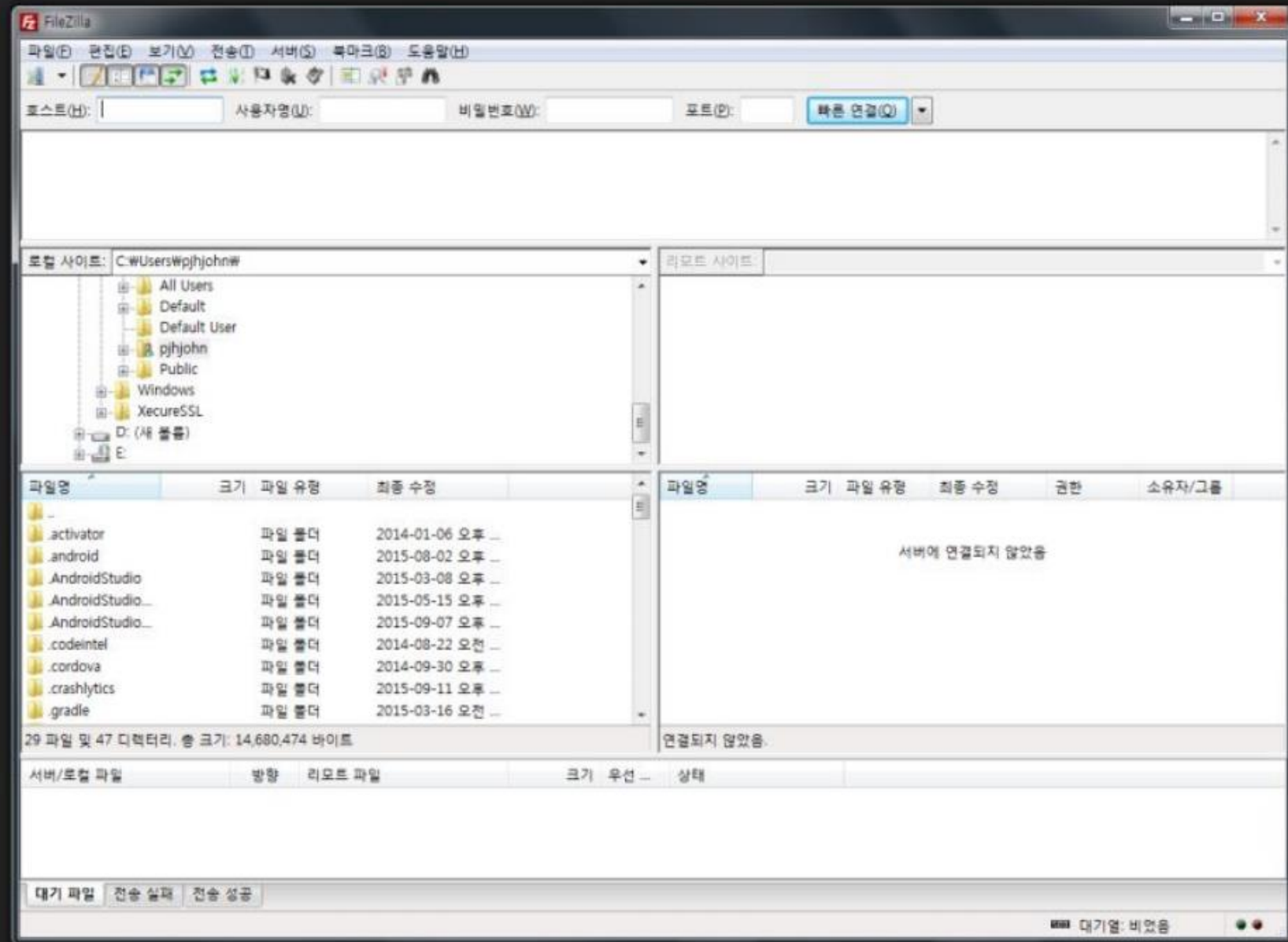
Click

**Download
FileZilla Server**

Windows only

Pick the client if you want to transfer files. Get the server if you want to make files available for others.

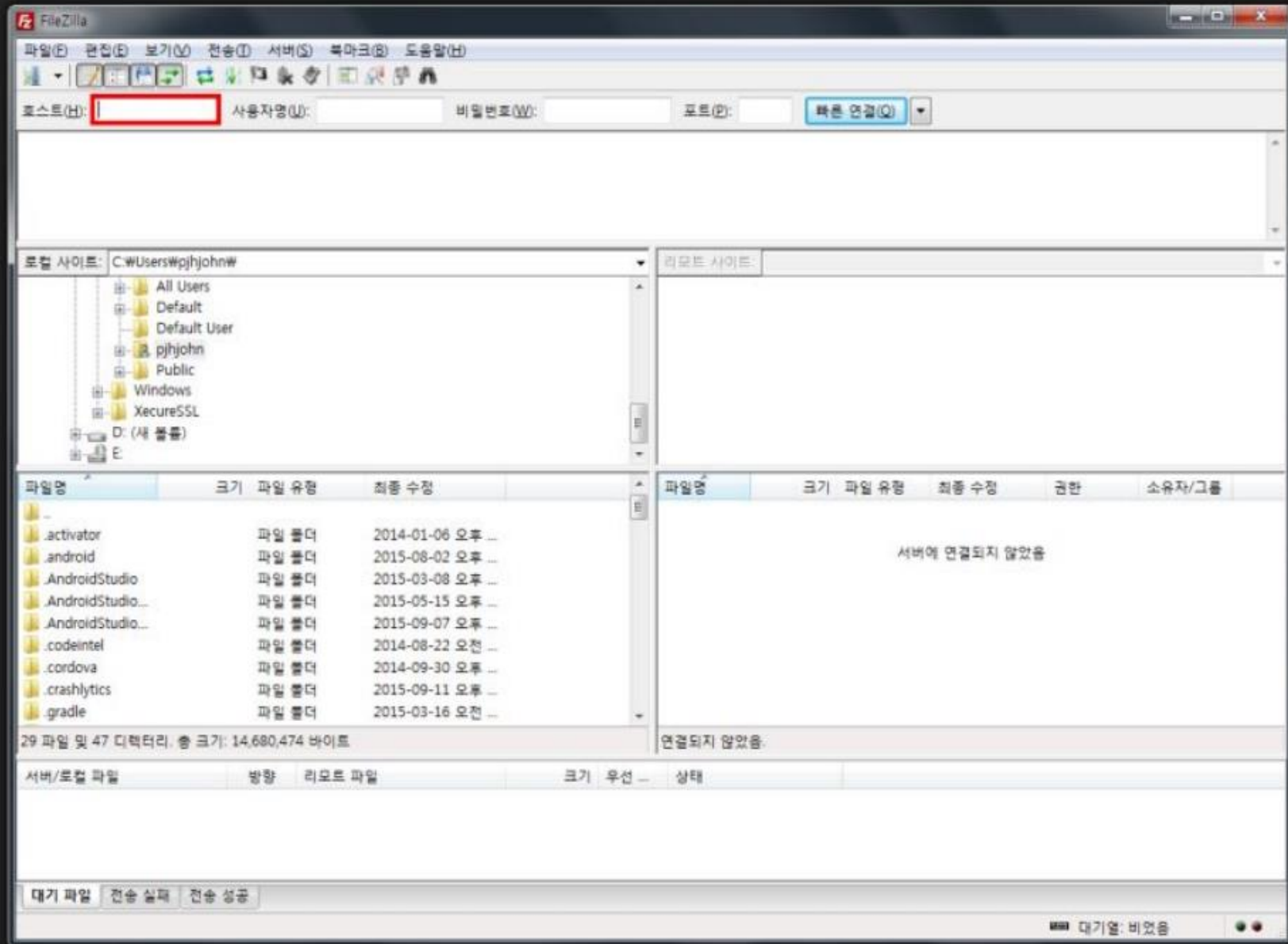
 **News** 





[호스트]

martini.snucse.org



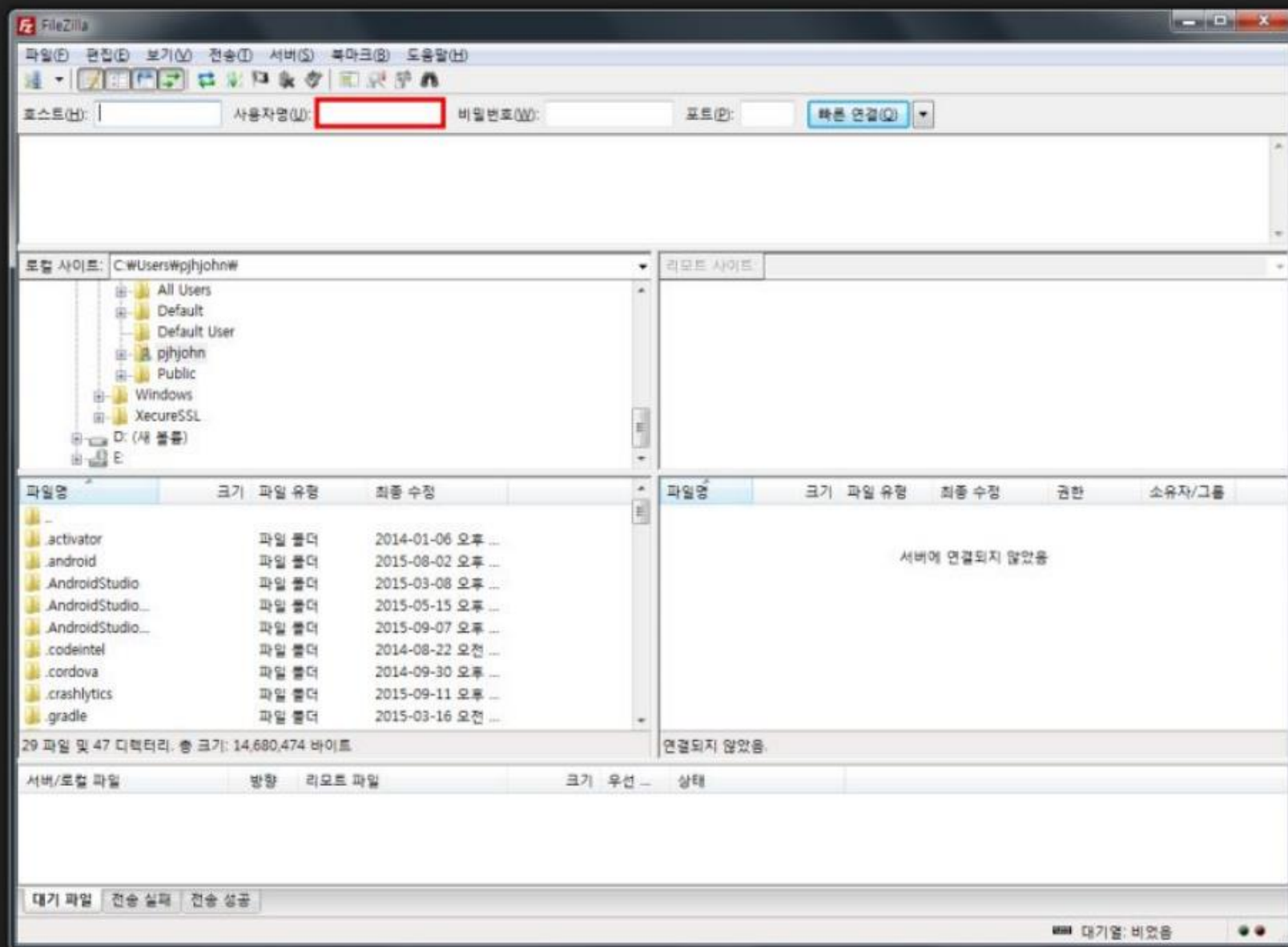


[호스트]

martini.snucse.org

[사용자명]

Your ID





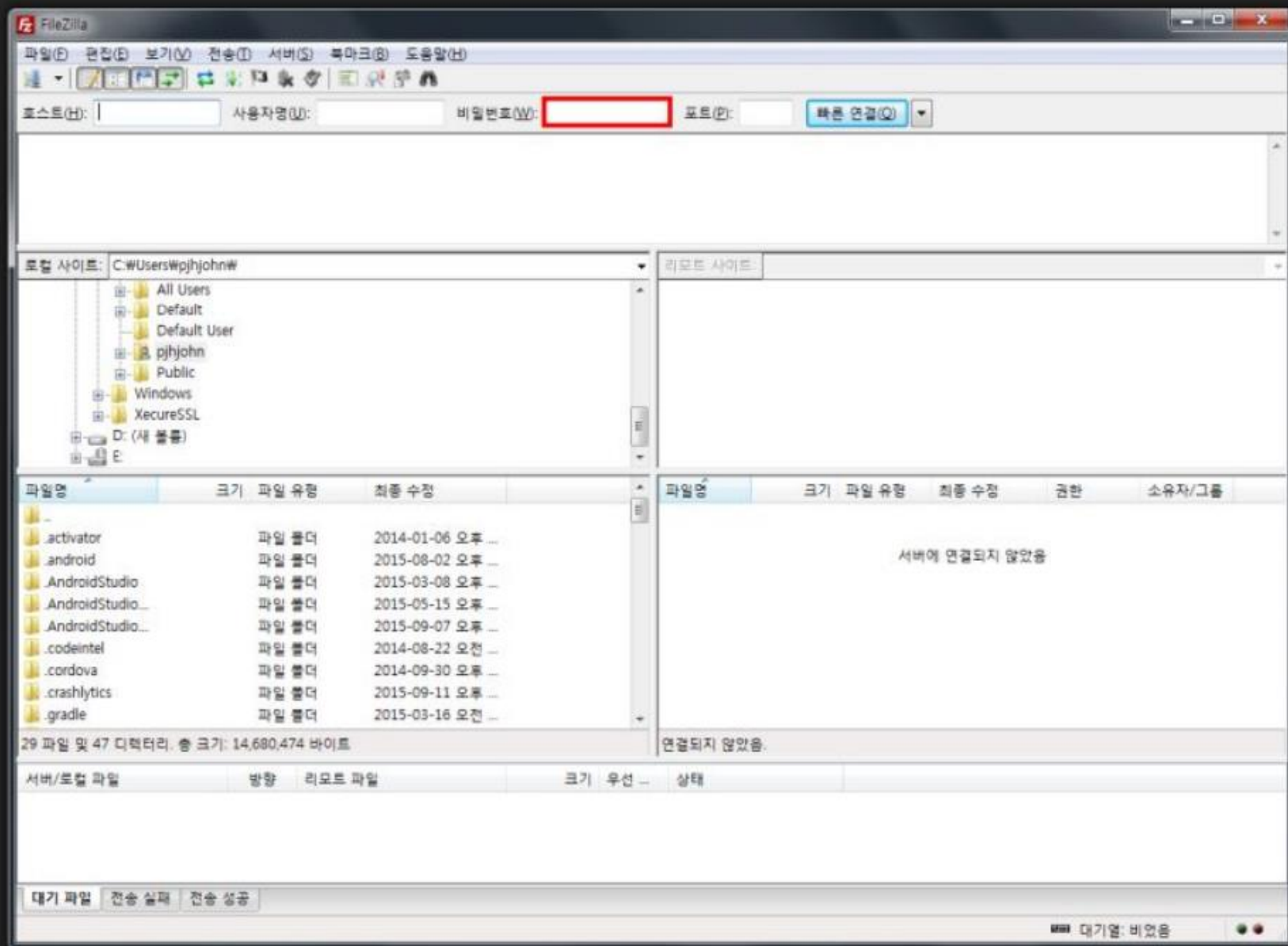
[호스트]

martini.snucse.org

[사용자명]

Your ID

[비밀번호]



[호스트]

martini.snucse.org

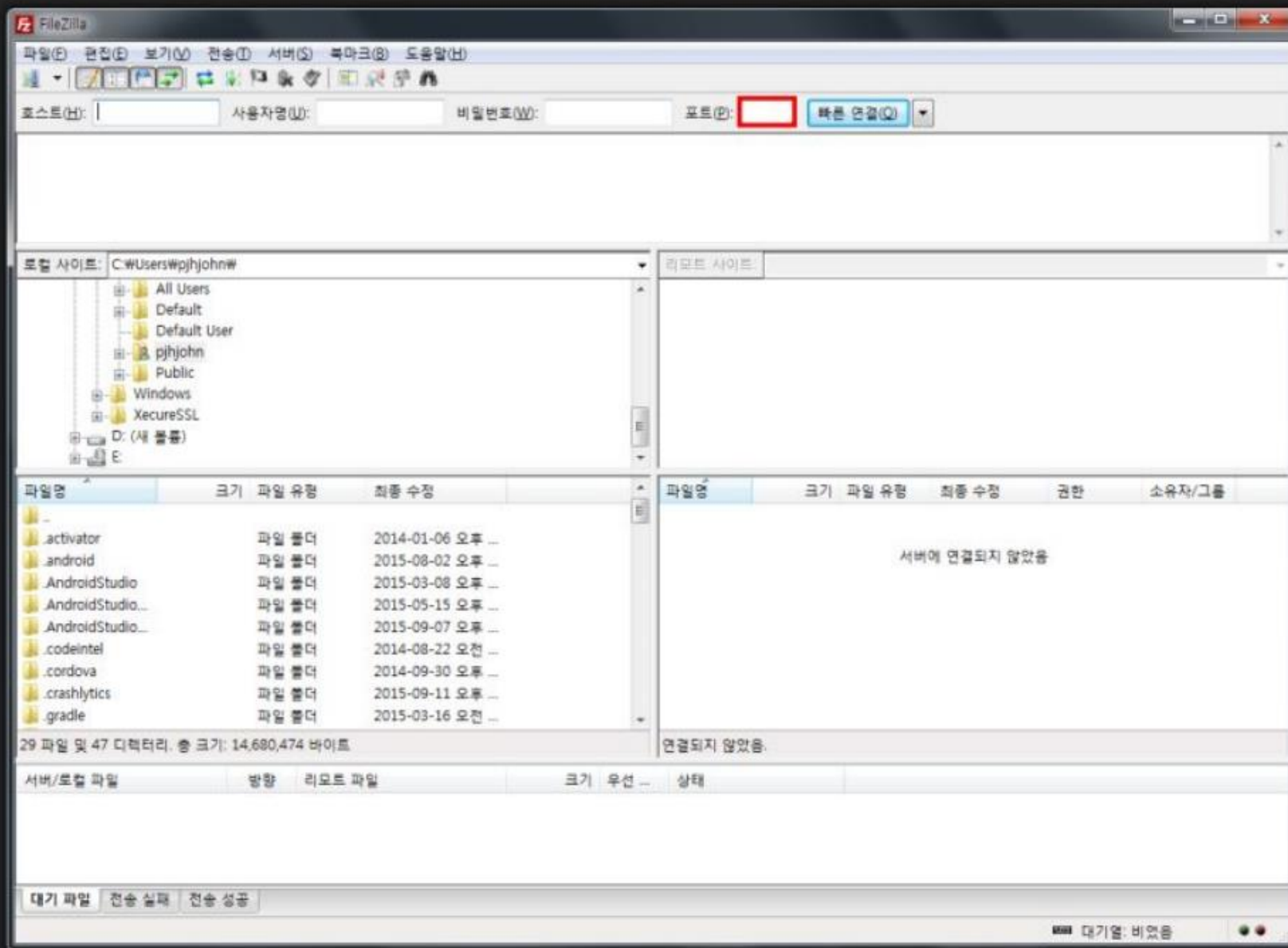
[사용자명]

Your ID

[비밀번호]

[포트]

22





[호스트]

martini.snucse.org

[사용자명]

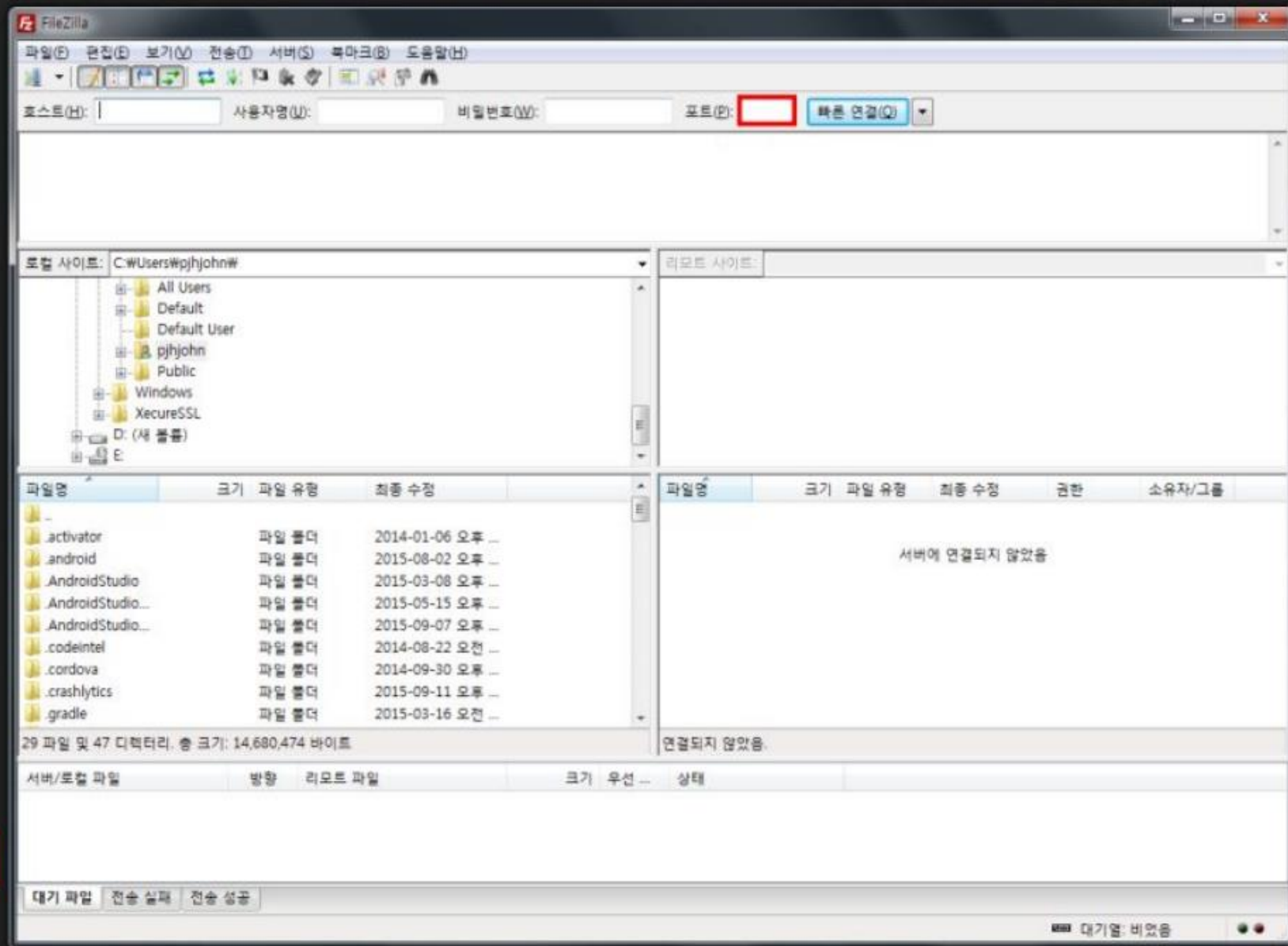
Your ID

[비밀번호]

[포트]

22

비밀번호 저장하지 않기





Drag & Drop 으로
파일 전송 가능

로그아웃 필수

