

SSH

Secure Shell Protocol의 약자로 컴퓨터와 컴퓨터가 인터넷과 같은 네트워크를 통해 서로 통신을 할 때 서로 안전하게 통신을 하기 위해 사용하는 프로토콜.

기존에는 Telnet이라는 프로토콜을 사용했으나 보안적으로 취약한 점이 있어서 SSH 프로토콜을 많이 사용하는 추세라고 함.

SSH 프로토콜을 사용할 때 일반 문자열을 출력하면 식별할 수 없는 내용으로 패킷이 보내지는데 이는 암호화된 통신을 가능하게 해 정보를 쉽게 탈취하지 못하고 또한 쉽게 해석하지 못하게 보안성을 향상시킬 수 있는 성질을 가지고 있음

Telnet

Telnet을 이용하면 네트워크에 있는 컴퓨터를 자유롭게 파일 전송, 파일 생성, 디렉토리 생성을 할 수 있다.

Telnet의 보안문제가 가장 큰데 이는 암호화 및 무결성 검사를 지원하지 않고, 패스워드가 암호화되어 있지 않아 제3자에게 노출될 수 있다.

Telnet은 정보를 byte 스트림 형식으로 주고받는 반면 SSH는 DES, RSA 등 고급 암호화를 통해 통신을 한다는 차이점이 있다.

HTTP

HyperText Transfer Protocol의 약자로 Hypertext는 컴퓨터 화면에서 볼 수 있는 텍스트이고, 다른 텍스트와 연결될 수 있는 주소를 참조하고 있다.

사람들이 브라우저를 통해 확인하는 인터넷 상의 데이터는 HTTP에 의해 전달되는 것

브라우저 창에 URL을 입력하고 그곳으로 들어가지는 것이 HTTP의 역할이며 이것을 HTTP Request, HTTP Response라고 한다.

HTTPS

HyperText Transfer Protocol Secure의 약자로 HTTP의 확장 버전이다. Secure의 말에서 알 수 있듯 데이터의 보안을 위한 조치가 추가되었으며 이때 SSL이 사용된다.

Secure Sockets Layer를 통해 데이터를 한쪽에서 다른 한 쪽으로 안전하게 보낼 수 있게 한다.

FTP

인터넷을 통한 파일 송수신만을 위해 고안된 프로토콜이기에 동작방식이 단순하다.

www 방식보다 파일을 빠른 속도로 한꺼번에 주고 받을 수 있는 장점이 있음

sFTP

일반적인 FTP와 다르게 PC와 서버 간의 데이터 전송을 암호화하기 때문에 해킹이나 보안상의 문제점을 방지하고 안전한 서비스 이용을 가능하게 한다.

FTP와는 완전히 관계가 없는 파일 전송 프로토콜이며 SSL을 이용한다.

SSH 연결 위에 일반 FTP로 접속을 하지만 실제 데이터는 암호화 되지 않는다.

SMTP

Simple Mail Transfer Protocol의 약자로 e-mail을 컴퓨터에서 다른 컴퓨터로 전송할 때 사용하는 메일 서버의 기본 프로토콜

POP3

Post Office Protocol 3는 이메일 공급업체 서버에서 로컬장치로 이메일을 내려받도록 되어 있음.

서버에서 동기화가 이루어지지 않으므로 메일서버에 있는 메일을 다운로드해서 가져온다는 의미

대다수의 POP는 로컬장치에 이메일을 내려받으면 서버에서는 이메일을 지우도록 기본 설정이 되어있어 중요한 정보가 있는 메일을 다운받고 메일 클라이언트가 삭제된다면 중요한 메일이 없어질 수도 있다.

POP3를 사용하면 전자 메일을 다운하고 동일한 컴퓨터를 사용하여 액세스할 수 있음

IMAP

Internet Message Access Protocol은 중앙서버에서 동기화가 이루어지기 때문에 모든 장치에서 동일한 이메일 폴더를 확인할 수 있다.

스마트폰, 태블릿, PC 모두 동일한 받은 편지함, 보낸 편지함, 이메일 메시지를 보게 돼서 모든 장치에서 동일하게 확인이 가능함.

IMAP를 사용해서 전자 메일 서비스를 읽으면 실제로 컴퓨터에 파일을 다운로드 하거나 저장되지 않음.

IMAP은 메시지를 클릭하면 메시지를 다운로드 하고 첨부파일은 자동으로 다운되지 않음

NNTP

뉴스 서버 간에 유즈넷 뉴스 기사를 전송하고 최종 사용자 클라이언트 애플리케이션에 의해 기사를 구독, 게시할 수 있게 하기 위한 애플리케이션 프로토콜의 하나다.

NNTP 클라이언트는 인터넷 익스플로러, 또 다른 웹브라우저의 일부로서 포함될 수 있으며 뉴스리더라고 불리는 별도의 클라이언트 프로그램을 사용할 수 있다.

DNS

Domain Name System은 인터넷 전화번호부.

DNS는 브라우저가 인터넷 소스를 로드할 수 있도록 도메인 이름을 IP주소로 반환한다.

Ex) www.naver.com -> 125.209.222.141 처럼 IP주소로 반환한다.

Shell

리눅스의 셸은 명령어와 프로그램을 실행할 때 사용하는 인터페이스

커널(kernel)과 사용자간의 다리역할을 하는 것으로 사용자로부터 명령을 받아 그것을 해석하고 프로그램을 실행하는 역할

Login

ID와 비밀번호를 입력해서 shell을 실행하는 것을 의미함

Ssh로 접속하거나 로컬에서 GUI 로그인 화면을 통해 shell을 실행하는 것

Logout

Logout은 현재까지 시스템에서 사용 중이던 모든 작업을 종료하고 시스템을 빠져 나가는 명령어

단순히 시스템을 빠져나가는 것이 아니라 현재까지 사용 중이던 모든 자원을 커널에 반납하고 부여받았던 모든 권한을 종료한다는 의미까지 포함되어 있음

Username

Linux 사용자 계정 이름

Password

Linux 사용자 계정 비밀번호

Prompt

컴퓨터가 사용자의 입력을 받을 준비가 되어 있다는 것을 사용자에게 알려주기 위해 화면에 나타내고 있는 신호.

Directory

파일을 보관하는 곳

디렉토리는 최상위에 해당되는 루트(/)를 중심으로 하위 디렉토리에 다수의 디렉토리가 존재하는 형태의 트리구조를 갖추고 있다.

File

리눅스 시스템은 디렉토리, 일반파일, 특수파일 등 세 종류의 파일을 가지는데 사용자가 이용할 수 있는 데이터의 실체를 의미하며 시스템이나 응용프로그램도 파일로 본다.

Symbolic Link

원본파일을 가리키도록 링크만 연결시킨 것으로 윈도우에서의 바로가기 파일이라고 생각하면 쉬움.

원본파일만 가리키기만 하고 있으므로 원본파일의 크기와 무관하며 원본파일이 삭제되어 존재하지 않을 경우에 빨간색으로 깜빡거리며 링크파일의 원본파일 없다는 것을 알려줌

Hard Link

원본 파일과 동일한 inode를 가지고 있기 때문에 원본 파일이 삭제 되더라도 inode를 가지고 있는 링크 파일은 여전히 사용 가능하다.

Root Directory

루트 디렉토리는 계층 구조에서 가장 높은 디렉토리 입니다.

드라이브나 폴더에 있는 다른 모든 폴더를 포함하며 파일까지 포함할 수 있습니다.

루트가 맨 위에 있고 하위 폴더가 아래로 떨어지는 거꾸로 된 트리를 사용하여 이를 시각화 할 수 있습니다.

Home directory

시스템의 사용자에게 주어진 파일들을 포함한 다중 사용자 운영 체제의 파일 시스템 디렉토리이다.

Absolute Path

절대 경로는 루트 디렉토리를 포함한 주소를 갖는 경로를 말한다.

컴퓨터 상의 디렉토리에서는 C:|를 항상 포함하며, URL에서는 항상 <http://>로 시작된다.

Relative Path

상대경로는 루트 디렉토리를 포함하지 않는 주소를 갖는 경로를 말한다.

현재 참조하고 있는 문서가 항상 기준이며 이 문서를 통해 상위 폴더와 하위 폴더 또는 현재 폴더를 연결한다.

서버 주소가 달라지더라도 이전 서버에서의 디렉토리 구조만 같다면 경로를 수정하지 않아도 된다.

date명령어

```
s21800201@walab-HGU:~$ date
2021. 03. 27. (토) 13:06:28 KST
s21800201@walab-HGU:~$ date
2021. 03. 27. (토) 13:06:32 KST
```

➔ 현재 시각을 확인하는 명령어!!!!

cal명령어

```
s21800201@walab-HGU:~$ cal
      3월 2021
일 월 화 수 목 금 토
 1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

➔ 년도, 월, 일을 달력 형태로 보여주는 명령어!!!

df명령어

```
s21800201@walab-HGU:~$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
udev                  16239464         0   16239464    0% /dev
tmpfs                 3252932         0    3252932    0% /run
/dev/mapper/ubuntu--vg-root 458936456 27515320 408038712    7% /
tmpfs                16264644         0   16264644    0% /dev/shm
tmpfs                 5120            0      5120    0% /run/lock
tmpfs                16264644         0   16264644    0% /sys/fs/cgroup
/dev/loop0             63616           63616    0 100% /snap/gtk-common-themes/1506
/dev/loop1            224256          224256    0 100% /snap/gnome-3-34-1804/66
/dev/loop2            223232          223232    0 100% /snap/gnome-3-34-1804/60
/dev/loop3             1024            1024    0 100% /snap/gnome-logs/100
/dev/loop6             384             384    0 100% /snap/gnome-characters/570
/dev/loop5            2560            2560    0 100% /snap/gnome-calculator/826
/dev/sda1             523248          6196   517052    2% /boot/efi
/dev/loop8            2304            2304    0 100% /snap/gnome-system-monitor/148
/dev/loop9             384             384    0 100% /snap/gnome-characters/550
/dev/loop11           56832           56832    0 100% /snap/core18/1944
/dev/loop12           66432           66432    0 100% /snap/gtk-common-themes/1514
tmpfs                 3252928         28    3252900    1% /run/user/121
/dev/sdb1             5813266968      107468 5520117068    1% /home2
/dev/sdc1             5813266968      90140 5520134396    1% /backup
tmpfs                 3252928         0    3252928    0% /run/user/1000
/dev/loop14           56832           56832    0 100% /snap/core18/1998
/dev/loop4            2304            2304    0 100% /snap/gnome-system-monitor/157
/dev/loop15           640             640    0 100% /snap/gnome-logs/103
/dev/loop16           2560            2560    0 100% /snap/gnome-calculator/884
/dev/loop10           33152           33152    0 100% /snap/snapd/11107
/dev/loop7            33152           33152    0 100% /snap/snapd/11402
tmpfs                 3252928         0    3252928    0% /run/user/1036
tmpfs                 3252928         0    3252928    0% /run/user/1035
tmpfs                 3252928         0    3252928    0% /run/user/1027
tmpfs                 3252928         0    3252928    0% /run/user/1050
tmpfs                 3252928         0    3252928    0% /run/user/1019
```

➔ 하드디스크에 남은 용량을 확인할 때 사용하는 명령어!!!

free명령어

```
s21800201@walab-HGU:~$ free
              total        used        free      shared  buff/cache   available
Mem:         32529288      2139804      4241804        32476      26147680      29892132
Swap:         1003516           0       1003516
```

➔ 사용 중이 아닌 메모리 크기로 사용 가능한 메모리 크기

exit명령어

```
s21800201@walab-HGU:~$ exit
logout
Connection to walab.handong.edu closed.
```

→ 셸이나 프로그램을 종료할 때 쓰이는 명령어

pwd명령어

```
s21800201@walab-HGU:~$ pwd
/home2/class2021/s21800201
```

→ 현재 작업 중인 디렉토리의 절대 경로를 출력하는 명령어

cd명령어

```
s21800201@walab-HGU:~$ pwd
/home2/class2021/s21800201
s21800201@walab-HGU:~$ cd html/
s21800201@walab-HGU:~/html$ pwd
/home2/class2021/s21800201/html
```

→ Change directory 의 약자로 디렉토리를 변경할 때 사용되는 명령어

ls명령어

```
s21800201@walab-HGU:~$ ls
data.txt  data2.txt  error  examples.desktop  examples.desktop2  examples_link  html  list.txt  webapps
```

→ 현재 디렉토리 파일에 대한 리스트를 보여주는 명령어

file명령어

```
s21800201@walab-HGU:~$ file -C
s21800201@walab-HGU:~$ file --help
Usage: file [OPTION...] [FILE...]
Determine type of FILES.

  -h, --help                display this help and exit
  -v, --version             output version information and exit
  -m, --magic-file LIST    use LIST as a colon-separated list of magic
                           number files
  -z, --uncompress          try to look inside compressed files
  -Z, --uncompress-noreport only print the contents of compressed files
  -b, --brief               do not prepend filenames to output lines
  -c, --checking-printout   print the parsed form of the magic file, use in
                           conjunction with -m to debug a new magic file
                           before installing it
  -e, --exclude TEST       exclude TEST from the list of test to be
                           performed for file. Valid tests are:
                           apptype, ascii, cdf, compress, elf, encoding,
                           soft, tar, text, tokens
  -f, --files-from FILE    read the filenames to be examined from FILE
  -F, --separator STRING   use string as separator instead of ':'
  -i, --mime               output MIME type strings (--mime-type and
                           --mime-encoding)
  --apple                 output the Apple CREATOR/TYPE
  --extension             output a slash-separated list of extensions
  --mime-type             output the MIME type
  --mime-encoding         output the MIME encoding
  -k, --keep-going         don't stop at the first match
  -l, --list               list magic strength
  -L, --dereference        follow symlinks (default if POSIXLY_CORRECT is set)
  -h, --no-dereference     don't follow symlinks (default if POSIXLY_CORRECT is not set)
  -n, --no-buffer          do not buffer output
  -N, --no-pad             do not pad output
```

➔ 파일의 종류를 확인하는 명령어

less명령어

```
s21800201@walab-HGU:~$ less list.txt
```

```
total 64
drwx----- 6 s21800201 s21800201 4096 3월 26 10:31 .
drwxr-xr-x 90 root      root      4096 3월 18 22:19 ..
-rw----- 1 s21800201 s21800201 269 3월 23 11:15 .bash_history
-rw-r--r-- 1 s21800201 s21800201 220 4월 5 2018 .bash_logout
-rw-r--r-- 1 s21800201 s21800201 3771 4월 5 2018 .bashrc
drwx----- 2 s21800201 s21800201 4096 3월 23 10:42 .cache
drwx----- 3 s21800201 s21800201 4096 3월 23 10:42 .gnupg
-rw-r--r-- 1 s21800201 s21800201 807 4월 5 2018 .profile
-rw-r--r-- 2 s21800201 s21800201 8980 4월 16 2018 examples.desktop
-rw-r--r-- 2 s21800201 s21800201 8980 4월 16 2018 examples.desktop2
lrwxrwxrwx 1 s21800201 s21800201 16 3월 26 10:24 examples_link -> examples.desktop
drwxr-xr-x 2 s21800201 s21800201 4096 3월 18 21:42 html
-rw-rw-r-- 1 s21800201 s21800201 0 3월 26 10:32 list.txt
drwxr-xr-x 2 s21800201 s21800201 4096 2월 27 13:39 webapps
list.txt (END)
```

more명령어

```
s21800201@walab-HGU:~$ more list.txt
total 64
drwx----- 6 s21800201 s21800201 4096 3월 26 10:31 .
drwxr-xr-x 90 root      root      4096 3월 18 22:19 ..
-rw----- 1 s21800201 s21800201 269 3월 23 11:15 .bash_history
-rw-r--r-- 1 s21800201 s21800201 220 4월 5 2018 .bash_logout
-rw-r--r-- 1 s21800201 s21800201 3771 4월 5 2018 .bashrc
drwx----- 2 s21800201 s21800201 4096 3월 23 10:42 .cache
drwx----- 3 s21800201 s21800201 4096 3월 23 10:42 .gnupg
-rw-r--r-- 1 s21800201 s21800201 807 4월 5 2018 .profile
-rw-r--r-- 2 s21800201 s21800201 8980 4월 16 2018 examples.desktop
-rw-r--r-- 2 s21800201 s21800201 8980 4월 16 2018 examples.desktop2
lrwxrwxrwx 1 s21800201 s21800201 16 3월 26 10:24 examples_link -> examples.desktop
drwxr-xr-x 2 s21800201 s21800201 4096 3월 18 21:42 html
-rw-rw-r-- 1 s21800201 s21800201 0 3월 26 10:32 list.txt
drwxr-xr-x 2 s21800201 s21800201 4096 2월 27 13:39 webapps
```

➔ 긴 파일을 한 페이지씩 보여주는 명령어

cp명령어

```
s21800201@walab-HGU:~$ ls
data.txt data2.txt error examples.desktop examples
s21800201@walab-HGU:~$ cp data.txt data3.txt
s21800201@walab-HGU:~$ ls
data.txt data2.txt data3.txt error examples.desktop
```

- ➔ Data.txt를 복사해서 data3.txt을 생성한다
- ➔ Cp data.txt data3.txt dir3/ 하면 한번에 여러 개의 파일을 dir3에 저장할 수 있다

mv명령어

```
s21800201@walab-HGU:~$ ls
data.txt data2.txt data3.txt error examples.desktop
s21800201@walab-HGU:~$ mv data.txt data_moved.txt
s21800201@walab-HGU:~$ ls
data2.txt data3.txt data_moved.txt error examples.desktop
```

- ➔ Data.txt.를 data_moved.txt라는 이름으로 바꿔서 파일을 저장한다.

mkdir명령어

```
s21800201@walab-HGU:~$ mkdir Test
s21800201@walab-HGU:~$ ls -al
```

```
drwxrwxr-x  2 s21800201 s21800201 4096  3월 27 13:50 Test
```

- ➔ Test라는 디렉토리를 만드는 명령어

rm명령어

```
s21800201@walab-HGU:~$ rm data_moved.txt
Test data3.txt data_moved.txt error examples.desktop examples.desktop2
s21800201@walab-HGU:~$ rm data_moved.txt
s21800201@walab-HGU:~$ ls
Test data3.txt error examples.desktop examples.desktop2 examples_link
```

- ➔ Rm data_moved.txt는data_moved.txt를 삭제한다.

ln명령어

```
s21800201@walab-HGU:~$ ln data3.txt data_link
s21800201@walab-HGU:~$ ls
Test      data_link  examples.desktop  examples_link  list.txt  webapps
data3.txt error      examples.desktop2 html           magic.mgc
```

➔ Data3.txt의 하드링크를data_link로 생성하는 명령어(예시)

```
s21800201@walab-HGU:~$ ln -sf data3.txt data_sf_link
s21800201@walab-HGU:~$ ls
Test      data_link  error      examples.desktop2  html      magic.mgc
data3.txt data_sf_link examples.desktop  examples_link      list.txt  webapps
```

➔ Data3.txt의 심볼릭 링크를 data_sf_link로 생성하는 명령어(예시)

type명령어

```
s21800201@walab-HGU:~$ ls
Test      data3.txt  data_sf_link  examples.desktop  examples_link  list.txt  webapps
data2.txt data_link  error        examples.desktop2  html           magic.mgc
s21800201@walab-HGU:~$ type data3.txt data2.txt > data4.txt
-bash: type: data3.txt: not found
-bash: type: data2.txt: not found
```

➔ 원하는 파일을 지정에 하나의 파일로 병합하는 명령어

➔ Data3.txt: not found라는 커맨드가 뜬.

which명령어

```
s21800201@walab-HGU:~$ which ls
/bin/ls
s21800201@walab-HGU:~$ which date
/bin/date
s21800201@walab-HGU:~$ which cal
/usr/bin/cal
```

➔ 명령어의 위치를 알아내는 명령어!!

help명령어

```
s21800201@walab-HGU:~$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
-m, --mode=MODE    set file mode (as in chmod), not a=rwx - umask
-p, --parents       no error if existing, make parent directories as needed
-v, --verbose       print a message for each created directory
-Z, --context=CTX   set SELinux security context of each created directory to the default type
                   or SMACK security context to CTX
--context[=CTX]    like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX
--help             display this help and exit
--version          output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Report mkdir translation bugs to <http://translationproject.org/team/>
Full documentation at: <http://www.gnu.org/software/coreutils/mkdir>
or available locally via: info '(coreutils) mkdir invocation'
```

➔ 명령어의 도움말을 볼 수 있게 해주는 명령어

man명령어

```
s21800201@walab-HGU:~$ man ls
```

```
LS(1)                                User Commands                                LS
NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none
  -ctuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE
```

➔ 온라인 매뉴얼을 볼 수 있게 해주는 명령어

apropos명령어

```
s21800201@walab-HGU:~$ apropos data
accessdb (8) - dumps the content of a man-db database in a human readable format
appres (1) - list X application resource database
appstreamcli (1) - Handle AppStream metadata and the AppStream index
base32 (1) - base32 encode/decode data and print to standard output
base64 (1) - base64 encode/decode data and print to standard output
bzip2recover (1) - recovers data from damaged bzip2 files
Clone (3pm) - recursively copy Perl datatypes
Data::Dump (3pm) - Pretty printing of data structures
Data::Dump::Filtered (3pm) - Pretty printing with filtering
Data::Dump::Trace (3pm) - Helpers to trace function and method calls
DBD::mysql (3pm) - MySQL driver for the Perl5 Database Interface (DBI)
DBD::Sponge (3pm) - Create a DBI statement handle from Perl data
```

➔ Data와 관련있는 명령어를 설명과 함께 출력해주는 명령어

info명령어

```
s21800201@wlab-HGU:~$ info pwd
```

```
Next: stty invocation, Up: Working context
19.1 'pwd' : Print working directory
=====
'pwd' prints the name of the current directory. Synopsis:
  pwd [OPTION]...

The program accepts the following options. Also see \*note Common options::.

'-L'
  '--logical'
  If the contents of the environment variable 'PWD' provide an
  absolute name of the current directory with no '.' or '..'
  components, but possibly with symbolic links, then output those
  contents. Otherwise, fall back to default '-P' handling.

'-P'
  '--physical'
  Print a fully resolved name for the current directory. That is,
  all components of the printed name will be actual directory
  names-none will be symbolic links.

If '-L' and '-P' are both given, the last one takes precedence. If
neither option is given, then this implementation uses '-P' as the
```

➔ 리눅스 명령어의 사용 방법, 옵션 등을 나타내는 명령어

whatis명령어

```
s21800201@wlab-HGU:~$ whatis ls
ls (1)                - list directory contents
s21800201@wlab-HGU:~$ whatis pwd
pwd (1)               - print name of current/working directory
s21800201@wlab-HGU:~$ whatis ln
ln (1)                - make links between files
```

➔ 명령어에 대한 기능을 간략하게 나타내는 명령어

alias명령어

```
s21800201@wlab-HGU:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo
s21800201@wlab-HGU:~$ alias cp='cpoy'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -aLF'
alias ls='ls --color=auto'
```

➔ 시스템에 설정되어 있는 alias를 확인하기 위한 명령어

➔ Alias 별칭=명령어 로 작성하면 긴 명령어를 별칭으로 저장하게 된다.

/home2/class2021/s21800201/2021OSS/lab4/2048_JAVA

README.md

pom.xml

screenshot.jpg

src/main/java/org/game/controller/GameMainWindow.java

model/BestScore.java

view/GameHint.java

GameKeyEvent.java

GameMatrix.java

GameNewCell.java

GameScore.java

resource/org/game/resources/game.xml

test/java/org/game/test/TestCountScore.java

CamScanner로 스캔하기

```
s21800201@walab-HGU:~/2021OSS/lab4$ cp 2048_JAVA/src/main/java/org/game/controller/GameMainWindow.java 2048_JAVA/javasrc/GameMainWindow.java
s21800201@walab-HGU:~/2021OSS/lab4$ cd 2048_JAVA/
s21800201@walab-HGU:~/2021OSS/lab4/2048_JAVA$ tree
```

```
tree
.
├── README.md
├── javasrc
│   └── GameMainWindow.java
├── pom.xml
├── screenshot.jpg
└── src
    ├── main
    │   ├── java
    │   │   └── org
    │   │       └── game
    │   │           ├── controller
    │   │           │   ├── GameMainWindow.java
    │   │           │   └── model
    │   │           │       └── BestScore.java
    │   └── resource
    │       └── org
    │           └── game
    │               └── resources
    │                   └── game.xml
    └── test
        ├── java
        │   └── org
        │       └── game
        │           └── test
        │               └── TestCountScore.java
        └── resource
```