

Chapter 12 원격 접속과 FTP

목차

01 텔넷과 SSH

02 파일 송수신

03 메일 송수신

학습목표



- 텔넷 서버를 설치하고 설정할 수 있다.
- 텔넷과 SSH를 이용해, 리눅스 시스템에 접속하여 작업할 수 있다.
- FTP 서버를 설치하고 설정할 수 있다.
- FTP 클라이언트로 FTP 서버에 접속하여 파일을 주고받을 수 있다.
- 메일 클라이언트를 이용하여 메일을 주고받을 수 있다.

00 Preview

00 Preview

■ 12장의 내용 구성

- 원격 접속 시 기본적으로 TCP/IP 프로토콜 중 5계층에 있는 프로토콜을 사용하는데, 대표적인 프로토콜로 텔넷과 SSH가 있음. 텔넷이나 SSH를 사용하여 원격으로 리눅스 시스템에 접속해서 작업을 수행할 수 있음
- 리눅스 시스템과 터미널로 연결하여 작업하는 것 외에 파일을 주고받기 위해 사용하는 FTP 프로토콜도 있음
- FTP로 리눅스와 리눅스 간, 리눅스와 윈도우 간에 파일을 주고받을 수 있음



01 텔넷과 SSH

01 텔넷과 SSH

■ 텔넷

- 텔넷: 원격으로 리눅스에 접속하는 프로그램의 이름이기도 하지만 원래는 프로토콜의 이름
TCP/IP 프로토콜 중 응용 계층에 있는 프로토콜 중 하나
- 텔넷을 사용하려면 텔넷 클라이언트와 텔넷 서버가 필요
- 텔넷 서비스를 제공하려면 텔넷 서버가 설치되어 있어야 하고, 텔넷으로 시스템에 접속하려면 텔넷 클라이언트가 있어야 함

```
[user1@localhost ~]$ rpm -qa | grep telnet  
telnet-0.17-85.el9.x86_64
```

- 설치된 패키지를 확인해 보니, 텔넷 클라이언트만 설치되어 있음

01 텔넷과 SSH

■ 텔넷 서버 설치하기

- 로키 리눅스에서는 텔넷 서버가 systemd 데몬에 의해 독자적으로 동작
- 텔넷 서버의 패키지 이름은 telnet-server
- dnf를 이용하여 telnet-server를 설치

```
[root@localhost ~]# dnf install telnet-server
Rocky Linux 9 - BaseOS                               4.2 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream                             5.3 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                               3.6 kB/s | 2.9 kB    00:00
종속성이 해결되었습니다.
(생략)
설치되었습니다:
  telnet-server-1:0.17-85.el9.x86_64

완료되었습니다!
```


01 텔넷과 SSH

■ 텔넷 서버 설치하기

- 만약 텔넷 클라이언트가 설치되어 있지 않다면 텔넷 클라이언트를 설치함
- 텔넷 클라이언트의 패키지 이름은 telnet
- 텔넷 서버는 systemctl 명령으로 활성화하여 바로 사용할 수 있음
- 활성화할 서비스 이름은 telnet.socket

```
[root@localhost ~]# systemctl start telnet.socket
[root@localhost ~]# systemctl status telnet.socket
● telnet.socket - Telnet Server Activation Socket
   Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; preset:
disabled)
   Active: active (listening) since Mon 2023-10-02 15:22:31 KST; 9s ago
(생략)
```

01 텔넷과 SSH

■ 텔넷 사용하기

- 텔넷으로 로컬 호스트에 접속하면 바로 접속을 허용함. telnet 0에서 0은 로컬 호스트를 의미
- 텔넷으로 접속한 상태에서 로그인하지 않고 종료하려면 Ctrl +d를 입력하면 됨
- 로그인 프롬프트에서 아무 입력 없이 60초가 지나면 자동으로 종료됨

```
[root@localhost ~]# telnet 0
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.

Kernel 5.14.0-284.11.1.el9_2.x86_64 on an x86_64
localhost login:
Password:

Login incorrect

localhost login: login: timed out after 60 secondsConnection closed by foreign host.
```

01 텔넷과 SSH

■ 텔넷 모드 사용하기

- 텔넷을 아무 인자 없이 사용하면 다음과 같이 > 프롬프트가 뜨면서 텔넷 모드로 전환됨

```
[root@localhost ~]# telnet  
telnet>
```

- 여기서 open 명령을 사용하여 리눅스 시스템에 접속하거나, quit 명령을 사용하여 텔넷을 종료할 수 있음

```
[root@localhost ~]# telnet  
telnet> open 192.168.147.129  
Trying 192.168.147.129...  
Connected to 192.168.147.129.  
Escape character is '^'.  
  
Kernel 5.14.0-284.11.1.el9_2.x86_64 on an x86_64  
localhost login:
```

01 텔넷과 SSH

■ 직접 서버로 접속하기

- 텔넷에 인자로 접속할 서버의 주소나 호스트 이름을 지정하면 바로 해당 시스템에 접속할 수 있음
- 로컬 호스트에 접속할 때는 telnet 0이나 telnet localhost를 사용할 수 있으며, 로컬 호스트의 IP 주소를 지정해도 됨

```
[root@localhost ~]# telnet localhost
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Kernel 5.14.0-284.11.1.el9_2.x86_64 on an x86_64
localhost login:
```

01 텔넷과 SSH

■ 윈도 텔넷 사용하기

- 윈도에서는 텔넷 클라이언트를 제공하지만, 기본으로 설정되어 있지는 않음
- 윈도에서 텔넷을 사용하려면 윈도 단축키 (Win)+R을 누르고, 실행 창에서 'optionalfeatures'를 입력한 후 Enter 를 누름
- 그러면 [그림 12-1]과 같은 Windows 기능 켜기/끄기 창이 뜨는데, 여기서 '텔넷 클라이언트'를 선택하고 [확인]을 클릭하면 텔넷 클라이언트가 활성화 됨

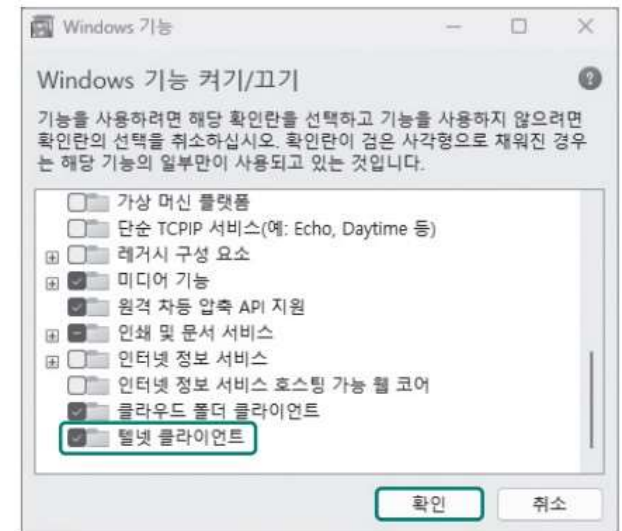


그림 12-1 Windows 기능 켜기/끄기 창

01 텔넷과 SSH

■ 윈도 텔넷 사용하기

- 윈도 텔넷 클라이언트는 [시작]-[실행]에서 telnet을 실행하면 [그림 12-2]와 같이 텔넷 모드의 상태로 텔넷 창이 뜬다. 여기서 open 명령으로 리눅스에 접속할 수 있음

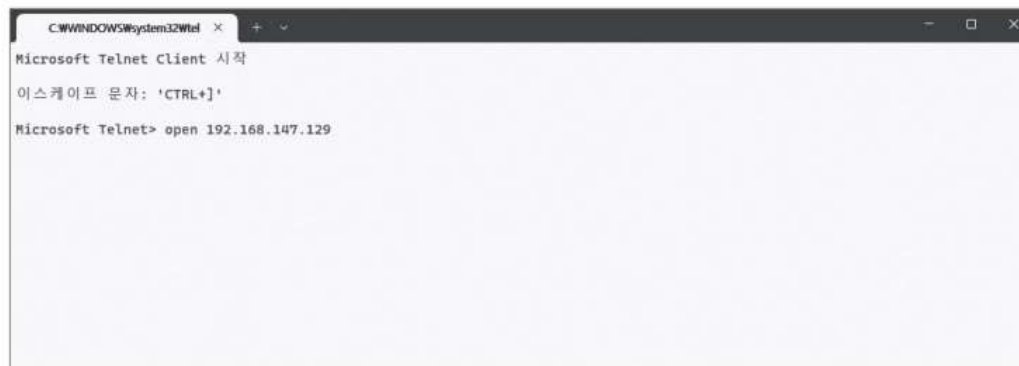


그림 12-2 윈도 텔넷으로 리눅스 접속

- 그런데 텔넷은 보안에 문제가 있기 때문에 리눅스의 방화벽에서 기본적으로 막아 놓음
- 따라서 [그림 12-2]와 같이 윈도 텔넷으로 접속하면 "연결 대상 192.168.147.129...호스트에 연결할 수 없습니다. 포트 23: 연결하지 못했습니다."라는 메시지가 출력됨

01 텔넷과 SSH

■ 윈도 텔넷 사용하기

- 방화벽에서 허용해야 외부에서 텔넷으로 접속할 수 있음
- 방화벽에 telnet을 허용하도록 설정하고 넘어가기. 이 작업은 root 계정에서 수행해야 함

```
[root@localhost ~]# firewall-cmd --add-service=telnet  
success
```

- 방화벽에 telnet을 추가하고 윈도 텔넷 창에서 다시 접속하면 [그림 12-3]과 같이 텔넷으로 리눅스에 접속할 수 있음



그림 12-3 윈도 텔넷으로 리눅스 접속 성공

01 텔넷과 SSH

■ 윈도 텔넷 사용하기

- 그런데 윈도의 한글과 리눅스 한글이 서로 맞지 않아 윈도 텔넷에서는 한글이 모두 깨져 보인다는 문제가 있음



```
Telnet 192.168.147.129 x + -
Kernel 5.14.0-284.11.1.el9_2.x86_64 on an x86_64
localhost login: user1
Password:
Last login: Mon Oct 2 15:16:17 from 192.168.147.1
[user1@localhost ~]$ ls
Test/ at.out nohup.out pw.dat temp/ tmp1/ tmp2/  총틀릿/ ?ㄷ승潘형뽕/ ㄷ뽕뽕/ ㄷ뽕뽕?뽕뽕/ ㄷ뽕뽕?? ?b뽕
/ ?뽕뽕/ ?뽕뽕/
[user1@localhost ~]$
```

그림 12-4 윈도 텔넷의 한글 깨짐 현상

01 텔넷과 SSH

■ PuTTY 사용하기

- PuTTY는 원격 접속 프로그램으로 문자 집합을 선택할 수 있어 윈도 텔넷처럼 한글이 깨져 보이는 현상을 해결할 수 있음
- PuTTY는 putty.org에서 다운로드할 수 있음. 가장 최신 버전은 2023년 10월 현재 0.79 버전
putty-64bit-0.79-installer.msi 파일을 다운로드하여 설치하면 됨
- PuTTY를 설치하고 putty.exe 파일을 실행하면 [그림 12-5]와 같은 설정 창이 나타남
- 여기서 호스트 이름이나 IP 주소를 입력하고 접속 형식으로 'Telnet'을 선택함

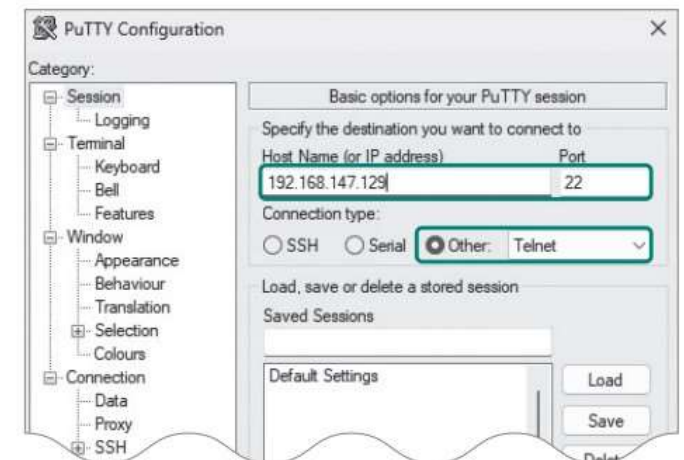


그림 12-5 PuTTY 접속 창

01 텔넷과 SSH

■ PuTTY 사용하기

- PuTTY는 기본 문자 집합을 UTF-8로 사용하므로 한글이 깨지지 않고 잘 보임



```
user1@localhost~  
Kernel 5.14.0-284.11.1.el9_2.x86_64 on an x86_64  
localhost login: user1  
Password:  
Last login: Mon Oct 2 15:45:55 from ::ffff:192.168.147.1  
[user1@localhost ~]$ ls  
Test/ at.out nohup.out pw.dat temp/ tmp1/ tmp2/ 글개 / 다운로드 / 문서 / 백담화면 / 비디오 / 사진 / 서식 / 음악 /  
[user1@localhost ~]$
```

그림 12-6 PuTTY로 텔넷 접속

01 텔넷과 SSH

■ SSH

- 텔넷은 원격으로 리눅스에 접속하는 대표적인 방법이지만 텔넷 클라이언트와 리눅스 사이에 주고받는 데이터가 암호화되지 않는다는 단점이 있음
- tcpdump 같은 패킷 캡처 프로그램을 사용하여 패킷을 캡처한다면 계정의 암호가 그대로 노출될 수 있음. 특히 root 계정의 암호가 노출될 수 있기 때문에 매우 위험함
- 이런 문제를 방지하기 위해 나온 프로토콜이 SSHSecure Shell

01 텔넷과 SSH

■ SSH

- SSH는 텔넷처럼 원격으로 시스템에 접속할 수 있도록 하지만 모든 통신을 암호화하여 주고 받음
- 로키 리눅스에서는 SSH 데몬이 기본으로 동작하며 ssh 클라이언트도 제공
- 만약 openssh가 설치되어 있지 않다면 직접 설치하고 활성화해야 함

```
[root@localhost ~]# rpm -qa | grep ssh
libssh-config-0.10.4-8.el9.noarch
libssh-0.10.4-8.el9.x86_64
openssh-8.7p1-29.el9_2.x86_64
openssh-clients-8.7p1-29.el9_2.x86_64
openssh-server-8.7p1-29.el9_2.x86_64
```

01 텔넷과 SSH

■ SSH

- SSH로 처음 접속하면 암호화 기법인 RSA를 사용한 인증 키를 생성할 것인지를 물어보는데, 이때 'yes'를 입력하면 접속됨
- 인증 키가 등록되면 이후 다시 접속할 때 SSH 인증 키를 생성할 것인지 묻지 않음

```
[root@localhost ~]# ssh 192.168.147.129
The authenticity of host '192.168.147.129 (192.168.147.129)' can't be established.
ED25519 key fingerprint is SHA256:S8xkSW2os27EdmNbMhQQDsAtBGlUCg+lXw9sKsD14+A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

01 텔넷과 SSH

■ SSH

- 윈도우에서 SSH로 리눅스에 접속하려면 PuTTY를 사용하면 됨
- [그림 12-7]과 같이 PuTTY 설정 창에서 접속 형식으로 'SSH'를 선택

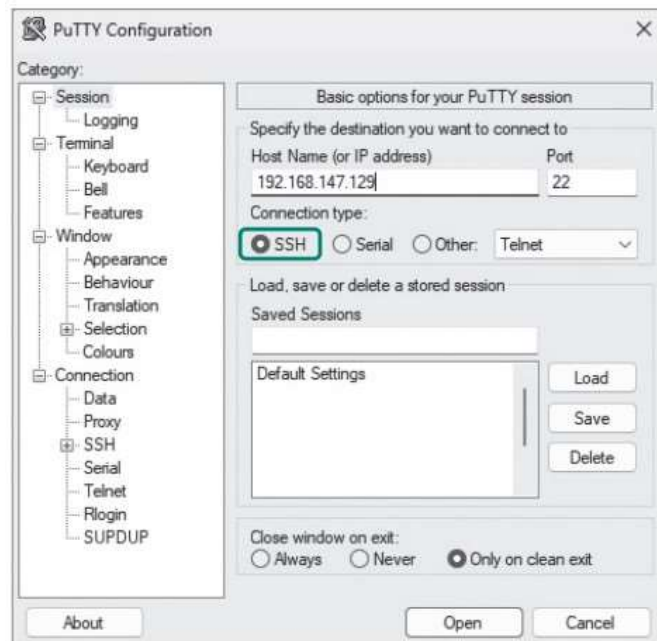


그림 12-7 HPuTTY로 SSH 접속

02 파일 송수신

02 파일 송수신

■ FTP

- 파일을 송수신할 때 사용하는 대표적인 방법은 FTP. FTP는 TCP/IP 프로토콜 중 5계층에 있는 프로토콜로, 리눅스뿐 아니라 이 프로토콜에 따르는 다른 운영체제와도 파일을 주고받을 수 있음
- 리눅스에서 사용하는 대표적인 FTP 서버는 vsFTPD

■ FTP 서버 설치하기

- 로키 리눅스에서 FTP 서버는 기본으로 설치되어 있지 않으므로 dnf 명령으로 설치해야 함

```
[root@localhost ~]# rpm -qa | grep ftp
[root@localhost ~]# dnf install vsftpd
(생략)
설치되었습니다:
  vsftpd-3.0.5-4.el9.x86_64

완료되었습니다!
```


02 파일 송수신

■ FTP 서버 설정하고 동작시키기

- vsFTPD의 설정은 /etc/vsftpd/vsftpd.conf 파일에서 진행

```
[root@localhost ~]# more /etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
(생략)
```

02 파일 송수신

■ FTP 서버 설정하고 동작시키기

- vsFTPD는 systemd 데몬으로 관리. vsftpd를 시작하거나 중지하려면 다음과 같이 함

```
[root@localhost ~]# systemctl stop vsftpd.service  
[root@localhost ~]# systemctl start vsftpd.service
```

- 리눅스에서 FTP 서비스를 제공하려면 방화벽에 신뢰할 수 있는 서비스로 FTP를 추가함

```
[root@localhost ~]# firewall-cmd --add-service=ftp  
success
```

- FTP 서버의 설치가 완료되었음. FTP 서버가 정상적으로 동작하는지 확인함
- FTP 클라이언트를 이용하여 접속할 수도 있지만 간단한 확인은 telnet 명령으로도 할 수 있음

02 파일 송수신

■ FTP 서버 설정하고 동작시키기

- 텔넷으로 FTP 포트 번호인 21을 연결하였을 때 '220 (vsFTPd 3.0.5)'라는 메시지가 출력 되면 FTP 서버가 제대로 동작하고 있는 것

```
[root@localhost ~]# telnet 0 21
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
220 (vsFTPd 3.0.5)
```

- FTP가 정상적으로 동작하는 것을 확인했으므로 quit로 종료

```
220 (vsFTPd 3.0.5)
quit
221 Goodbye.
Connection closed by foreign host.
[root@localhost ~]#
```

02 파일 송수신

■ FTP 클라이언트 설치하기

- FTP 클라이언트 패키지의 이름은 ftp. dnf 명령으로 ftp를 설치해 보자

```
[root@localhost ~]# dnf install ftp
(생략)
설치되었습니다:
  ftp-0.17-89.el9.x86_64

완료되었습니다!
```

- FTP 클라이언트가 동작하는지 확인. FTP 클라이언트의 명령 이름은 ftp. ftp가 정상적으로 동작하면 ftp> 프롬프트가 출력됨. ftp를 종료하는 명령은 bye

```
[root@localhost ~]# ftp
ftp> bye
[root@localhost ~]#
```

- 윈도에서 리눅스 서버와 FTP로 파일을 주고받을 때는 별도의 FTP 전용 프로그램(예.FileZilla)을 사용하는 것이 편리

02 파일 송수신

■ FTP 서버 접속하기

- FTP 클라이언트에서 FTP 서버에 접속하고 파일을 주고받는 방법을 알아보자
- FTP 클라이언트를 사용하려면 우선 FTP 서버의 호스트명(또는 IP 주소)과 로그인명, 암호를 알아야 함
- FTP는 기본적으로 두 컴퓨터 사이에 파일을 주고받을 때 사용하지만 여기서는 리눅스 한 대로 실습

- ① 터미널 2에서 user2 계정으로 로그인. 만약 user2 계정에 암호가 설정되어 있지 않다면 암호를 먼저 설정. 로그인하면 ftp 명령으로 리눅스 시스템에 접속함

```
[user2@localhost ~]$ ftp 192.168.147.129
Connected to 192.168.147.129 (192.168.147.129).
220 (vsFTPd 3.0.5)
Name (192.168.147.129:user2):
```

02 파일 송수신

■ FTP 서버 접속하기

- ② user1 계정으로 로그인. 만약 로그인명이나 암호를 잘못 입력하면 다음과 같은 메시지가 출력됨

```
[user2@localhost ~]$ ftp 192.168.147.129
Connected to 192.168.147.129 (192.168.147.129).
220 (vsFTPd 3.0.5)
Name (192.168.147.129:user2): user1
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp>
```

02 파일 송수신

■ FTP 서버 접속하기

- ③ 로그인명과 암호를 바르게 입력하면 로그인되었다는 메시지가 출력. 그리고 ftp 명령의 프롬프트인 ftp>가 출력됨

```
[user2@localhost ~]$ ftp 192.168.147.129
Connected to 192.168.147.129 (192.168.147.129).
220 (vsFTPd 3.0.5)
Name (192.168.147.129:user2): user1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

02 파일 송수신

■ FTP 내부 명령

- FTP 클라이언트에서 사용할 수 있는 명령

표 12-1 ftp의 내부 명령

내부 명령	의미
cd 원격 디렉터리	원격 호스트의 디렉터리를 이동한다.
lcd 지역 디렉터리	지역 호스트의 디렉터리를 이동한다.
pwd	원격 호스트의 디렉터리를 출력한다.
!pwd	지역 호스트의 디렉터리를 출력한다.
ls 또는 dir	원격 호스트의 파일 목록을 출력한다. dir 명령은 상세한 파일 정보를 출력한다.
!ls	지역 호스트의 파일 목록을 출력한다.
mkdir 원격 디렉터리	원격 호스트에 디렉터리를 생성한다.
rmdir 원격 디렉터리	원격 호스트의 디렉터리를 삭제한다.
get 원격 파일명 [지역 파일명]	원격 파일 하나를 지역 호스트로 가져온다. 지역 파일명을 지정하면 지정한 파일명으로 저장하고, 지정하지 않으면 원격 파일명과 동일한 파일명으로 저장한다.
mput 지역 파일명	여러 개의 지역 파일을 보낸다.
prompt	mget이나 mput 명령 사용 시 파일 전송 여부를 물어볼 것인지를 결정한다.
hash	파일이 전송되는 동안 #를 출력하여 진행 상황을 알려준다.
bin	바이너리 파일을 송수신할 것임을 지정한다.
bye	ftp를 종료한다.
open	ftp로 접속할 호스트를 입력하도록 한다.
user	사용자명을 다시 입력할 수 있도록 한다.
? 또는 help [명령]	명령에 대한 도움말을 출력한다.

02 파일 송수신

■ 파일 송수신하기

- ftp 명령으로 파일을 송수신할 때는 우선 dir이나 ls 명령으로 파일명을 확인하고 get, mget 명령으로 파일을 가져오거나 put, mput 명령으로 파일을 보냄
- get과 put 명령은 한 번에 파일 하나를 송수신하고, mget과 mput 명령은 한 번에 여러 개의 파일을 송수신할 수 있음
- mget이나 mput 명령을 사용하면 각 파일을 전송할 것인지 매번 물어보는데 prompt 명령으로 이를 설정하거나 해제할 수 있음

02 파일 송수신

■ pwd, dir 명령 사용하기

- 다음 예에서는 ftp로 리눅스 시스템에 접속한 후, pwd 명령으로 원격 시스템의 현재 디렉터리 위치를 확인하고 dir 명령으로 디렉터리의 내용을 확인

```
ftp> dir
227 Entering Passive Mode (192,168,147,129,60,43).
150 Here comes the directory listing.
drwxr-xr-x  8 1000 1000          72 Sep 03 11:11 Test
-rw-r--r--  1 1000 1000         846 Aug 26 12:00 at.out
-rw-----  1 1000 1000       130826 Aug 26 11:46 nohup.out
-rw-r--r--  1 1000 1000       65913 Aug 26 11:47 pw.dat
drwxr-xr-x  3 1000 1000          17 Aug 05 14:25 temp
drwxr-xr-x  2 1000 1000           6 Aug 05 14:23 tmp1
drwxr-xr-x  2 1000 1000           6 Aug 05 14:23 tmp2
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 공개
drwxr-xr-x  2 1000 1000       4096 Sep 03 06:51 다운로드
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 문서
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 바탕화면
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 비디오
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 사진
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 서식
drwxr-xr-x  2 1000 1000           6 Jul 09 01:48 음악
226 Directory send OK.
ftp>
```

02 파일 송수신

■ get 명령 사용하기

- get 명령을 사용하여 Test/ch4 디렉터리에 있는 test.txt 파일을 가져옴

```
ftp> cd Test/ch4
250 Directory successfully changed.
ftp> dir
227 Entering Passive Mode (192,168,147,129,137,17).
150 Here comes the directory listing.
-rw-r--r--  1 1000    1000          86 Aug 15 10:36 exec.txt
-rw-r--r--  1 1000    1000        146 Aug 15 12:03 exec2.txt
-rw-r--r--  1 1000    1000        169 Aug 15 12:08 exec3.txt
-rw-r--r--  1 1000    1000        290 Aug 15 11:59 test.txt
226 Directory send OK.
ftp> get test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (192,168,147,129,88,220).
150 Opening BINARY mode data connection for test.txt (290 bytes).
226 Transfer complete.
290 bytes received in 0.133 secs (2.19 Kbytes/sec)
ftp>
```

02 파일 송수신

■ mget 명령 사용하기

- 여러 파일을 주고받을 때는 mget 명령을 사용하는 것이 편리
- 예를 들어 exec.txt와 exec2.txt 파일을 한 번에 가져오려면 다음과 같이 함

```
ftp> mget exec.txt exec2.txt  
mget exec.txt?
```

02 파일 송수신

■ mget 명령 사용하기

- 그런데 mget 명령을 사용하면 각 파일을 수신할 것인지 물어봄
- 여기에 일일이 대답하기 귀찮으면 mget을 사용하기 전에 prompt 명령을 입력하면 됨
- 위의 예에서 n으로 대답한 뒤 다음 예제를 따라 해봄

```
ftp> mget exec.txt exec2.txt
mget exec.txt? n
mget exec2.txt? n
ftp> prompt
Interactive mode off.
ftp> mget exec.txt exec2.txt
local: exec.txt remote: exec.txt
227 Entering Passive Mode (192,168,147,129,228,57).
150 Opening BINARY mode data connection for exec.txt (86 bytes).
226 Transfer complete.
86 bytes received in 0.000678 secs (126.84 Kbytes/sec)
local: exec2.txt remote: exec2.txt
227 Entering Passive Mode (192,168,147,129,242,162).
150 Opening BINARY mode data connection for exec2.txt (146 bytes).
226 Transfer complete.
146 bytes received in 0.000647 secs (225.66 Kbytes/sec)
ftp>
```

02 파일 송수신

■ mget 명령 사용하기

- prompt 명령을 입력했더니 'Interactive mode off'가 되었다고 나옴
- Interactive mode(대화형 모드)란 사용자에게 여러 개의 파일을 송수신할 때마다 송수신할 것인지 물어보는 것을 뜻함
- 대화형 모드 끄기가 되었다는 것은 매번 물어보는 기능을 비활성화했다는 의미로, mget 명령을 사용할 때 수신 여부를 매번 물어보지 않고 바로 전송함

02 파일 송수신

■ hash 명령 사용하기

- ftp로 파일을 송수신할 때 진행 상황이 나타나도록 하려면 hash 명령을 사용
- hash 명령은 파일이 송수신 중이라는 표시로 #를 출력
- # 표시를 테스트하려면 파일의 크기가 큰 것이 좋음
- user1의 홈 디렉터리로 이동하여 pw.dat를 가져와 보자

```
ftp> pwd
257 "/home/user1" is the current directory
ftp> dir
227 Entering Passive Mode (192,168,147,129,255,42).
150 Here comes the directory listing.
drwxr-xr-x  8  1000  1000          72  Sep  03  11:11  Test
-rw-r--r--  1  1000  1000          846  Aug  26  12:00  at.out
-rw-----  1  1000  1000       130826  Aug  26  11:46  nohup.out
-rw-r--r--  1  1000  1000       65913  Aug  26  11:47  pw.dat
(생략)
226 Directory send OK.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> get pw.dat
local: pw.dat remote: pw.dat
227 Entering Passive Mode (192,168,147,129,171,159).
150 Opening BINARY mode data connection for pw.dat (65913 bytes).
#####
226 Transfer complete.
65913 bytes received in 0.107 secs (618.14 Kbytes/sec)
ftp>
```

02 파일 송수신

■ bye 명령 사용하기

- ftp 접속을 종료하는 명령은 bye

```
ftp> bye
221 Goodbye.
[user2@localhost ~]$
```


02 파일 송수신

■ 익명 FTP 사용 방법

- 익명 FTP: FTP 서비스를 제공하는 호스트에 등록된 로그인명을 몰라도 익명으로 ftp를 이용할 수 있도록 하는 것.
- 시스템 관리자가 익명 FTP 서비스를 제공하도록 설정해 놓은 호스트에게만 이 서비스를 제공할 수 있음. 이때 사용하는 로그인명은 anonymous이며, 암호에는 자신의 이메일 주소를 입력하면 됨
- 익명 FTP의 허용은 /etc/vsftpd/vsftpd.conf 파일에서 진행

```
[root@localhost ~]# vi /etc/vsftpd/vsftpd.conf
(생략)
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES    -> NO에서 YES로 수정한다.
(생략)
~
:wq
[root@localhost ~]# systemctl stop vsftpd
[root@localhost ~]# systemctl start vsftpd
```

02 파일 송수신

■ 익명 FTP 사용 방법

- 터미널 2에서 익명으로 접속
- 로그인 이름으로 anonymous를 사용하고, 암호는 적당한 이메일 주소를 입력

```
[user2@localhost ~]$ ftp 192.168.147.129
Connected to 192.168.147.129 (192.168.147.129).
220 (vsFTPd 3.0.5)
Name (192.168.147.129:user2): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

02 파일 송수신

■ 익명 FTP 사용 방법

- 익명으로 접속하여 파일을 가져오려면 서버에서 미리 파일을 준비해야 함

```
ftp> pwd
257 "/" is the current directory
ftp> dir
227 Entering Passive Mode (192,168,147,129,43,33).
150 Here comes the directory listing.
drwxr-xr-x  2 0      0              6 Apr 26 18:29 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> dir
227 Entering Passive Mode (192,168,147,129,143,99).
150 Here comes the directory listing.
226 Directory send OK.
ftp>
```

02 파일 송수신

■ FTP로 파일 보내고 받기

① 터미널 2에서 ftp로 리눅스 시스템에 접속. user1 계정을 사용

```
[user2@localhost ~]$ ftp 192.168.147.129
Connected to 192.168.147.129 (192.168.147.129).
220 (vsFTPd 3.0.5)
Name (192.168.147.129:user2): user1
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

02 파일 송수신

■ FTP로 파일 보내고 받기

② Test/ch8 디렉터리로 이동하고 디렉터리의 내용을 확인

```
ftp> cd Test/ch8
250 Directory successfully changed.
ftp> dir
227 Entering Passive Mode (192,168,147,129,135,137).
150 Here comes the directory listing.
drwxr-xr-x  3  1000  1000          73 Sep 03 11:14 ch2
-rw-r--r--  1  1000  1000      706560 Sep 03 11:18 ch2.tar
-rwxr-xr-x  1  1000  1000      25848 Sep 03 11:50 hello
-rw-r--r--  1  1000  1000        65 Sep 03 11:47 hello.c
-rw-r--r--  1  1000  1000       140 Sep 03 11:56 makefile
-rwxr-xr-x  1  1000  1000      25904 Sep 03 11:57 one
-rw-r--r--  1  1000  1000       135 Sep 03 11:53 one.c
-rw-r--r--  1  1000  1000       1640 Sep 03 11:57 one.o
-rw-r--r--  1  1000  1000     692252 Sep 03 11:18 services
-rw-r--r--  1  1000  1000       145 Sep 03 11:55 two.c
-rw-r--r--  1  1000  1000       1656 Sep 03 11:57 two.o
226 Directory send OK.
ftp>
```

02 파일 송수신

■ FTP로 파일 보내고 받기

- ③ prompt와 hash 명령을 사용하여 대화형 모드를 끄고 # 표시를 활성화

```
ftp> prompt
Interactive mode off.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp>
```

02 파일 송수신

■ FTP로 파일 보내고 받기

- ④ .c 파일을 가져옴. 파일의 크기가 작아 #가 하나씩 표시됨

```
ftp> mget *.c
local: hello.c remote: hello.c
227 Entering Passive Mode (192,168,147,129,101,131).
150 Opening BINARY mode data connection for hello.c (65 bytes).
#
226 Transfer complete.
65 bytes received in 0.37 secs (0.18 Kbytes/sec)
local: one.c remote: one.c
227 Entering Passive Mode (192,168,147,129,176,203).
150 Opening BINARY mode data connection for one.c (135 bytes).
#
226 Transfer complete.
135 bytes received in 0.0387 secs (3.49 Kbytes/sec)
local: two.c remote: two.c
227 Entering Passive Mode (192,168,147,129,120,187).
150 Opening BINARY mode data connection for two.c (145 bytes).
#
226 Transfer complete.
145 bytes received in 0.000926 secs (156.59 Kbytes/sec)
ftp>
```

02 파일 송수신

■ FTP로 파일 보내고 받기

⑤ user2 계정에 있는 pw.dat 파일을 업로드

```
ftp> put pw.dat
local: pw.dat remote: pw.dat
227 Entering Passive Mode (192,168,147,129,215,64).
150 Ok to send data.
#####
226 Transfer complete.
65913 bytes sent in 0.000173 secs (381000.00 Kbytes/sec)
ftp>
```

⑥ ftp를 종료

```
ftp> bye
221 Goodbye.
[user2@localhost ~]$
```


03 메일 송수신

03 메일 송수신

■ 메일 서버 설치 및 설정

- 리눅스에서 사용하는 기본 메일 서버는 sendmail
- dnf 명령으로 sendmail을 설치

```
[root@localhost ~]# dnf install sendmail
마지막 메타자료 만료확인(0:58:27 이전): 2023년 10월 02일 (월) 오후 07시 23분 45초.
종속성이 해결되었습니다.
(생략)
설치되었습니다:
  cyrus-sasl-2.1.27-21.el9.x86_64 openldap-compat-2.6.2-3.el9.x86_64 procmail-3.22-56.
el9.x86_64 sendmail-8.16.1-10.el9.x86_64
  tinycdb-0.78-18.el9.x86_64

완료되었습니다!
```

03 메일 송수신

■ 메일 서버 설정하고 동작시키기

- sendmail과 관련된 설정 파일은 /etc/mail/sendmail.mc
- 설정 파일을 수정하지 않고 메일이 동작하는지만 확인

```
[root@localhost ~]# more /etc/mail/sendmail.mc
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #      /etc/mail/make
dnl #
(생략)
```

03 메일 송수신

■ 메일 서버 설정하고 동작시키기

- sendmail을 사용하려면 systemd로 sendmail을 동작시키고 방화벽에서 25번 포트(SMTP)를 허용
- sendmail을 시작하거나 중지하려면 다음과 같이 함

```
[root@localhost ~]# systemctl stop sendmail.service  
[root@localhost ~]# systemctl start sendmail.service
```

- 리눅스에서 메일 서비스를 제공하려면 방화벽에 신뢰할 수 있는 서비스로 SMTP를 추가

```
[root@localhost ~]# firewall-cmd --add-service=smtp  
success
```

03 메일 송수신

■ 메일 서버 설정하고 동작시키기

- 메일 서버가 정상적으로 동작하는지 확인. FTP와 마찬가지로 telnet으로 확인
- 메일 서버의 포트 번호는 25번. 220으로 시작하는 메시지가 출력되면 메일 서버가 정상적으로 동작하고 있는 것

```
[root@localhost ~]# telnet 0 25
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
220 localhost.localdomain ESMTP Sendmail 8.16.1/8.16.1; Mon, 2 Oct 2023 20:26:03 +0900
```

- 메일이 정상적으로 동작하는 것을 확인했으므로 quit로 종료

```
220 localhost.localdomain ESMTP Sendmail 8.16.1/8.16.1; Mon, 2 Oct 2023 20:26:03 +0900
quit
221 2.0.0 localhost.localdomain closing connection
Connection closed by foreign host.
[root@localhost ~]#
```

03 메일 송수신

■ 메일 클라이언트 설치 및 설정

- 리눅스와 유닉스에서 전통적으로 사용하는 기본 메일 클라이언트는 mailx인데, CentOS Stream 9에서 mailx는 s-nail로 대체되었음
- 로키 리눅스의 경우에도 mailx가 설치되지 않은 버전을 사용한다면 s-nail 패키지를 설치

```
[root@localhost ~]# dnf install mailx
마지막 메타자료 만료확인(1:04:26 이전): 2023년 10월 02일 (월) 오후 07시 23분 45초.
일치하는 인수가 없습니다: mailx
오류: 일치하는 항목을 찾을 수 없습니다: mailx
[root@localhost ~]# dnf install s-nail
마지막 메타자료 만료확인(1:07:00 이전): 2023년 10월 02일 (월) 오후 07시 23분 45초.
종속성이 해결되었습니다.
설치되었습니다:
  s-nail-14.9.22-6.el9.x86_64

완료되었습니다!
```

03 메일 송수신

■ 메일 클라이언트 설치 및 설정

- mailx는 s-nail에 대한 심볼릭 링크로 제공

s-nail

- 기능 메일을 보내거나 읽는다.
- 형식 s-nail [옵션] [이메일 주소]
- 옵션 -s: 메일 제목 지정
- 사용 예 s-nail
s-nail user2@han.co.kr

03 메일 송수신

■ 메일 보내기

- 메일은 수신자의 주소와 제목, 본문 내용으로 구성됨. 수신자의 주소는 같은 시스템에 등록 된 내부 사용자라면 사용자명만 알면 되고, 다른 시스템에 있는 외부 사용자의 경우 '사용자 명@호스트명' 형태로 지정해야 함
- 내부 사용자에게 보내기 위한 메일 작성 예. user2 사용자가 user1 사용자에게 메일을 보내는 경우

```
[user2@localhost ~]$ s-nail user1
Subject: Hello
To: user1
mail test
bye
^D
-----
(Preliminary) Envelope contains:
To: user1
Subject: Hello
Send this message [yes/no, empty: recompose]? yes
```


03 메일 송수신

■ 메일 보내기

- user1 사용자에게 메일을 보내기 위해 's-nail user1'을 입력하면 바로 'Subject:'가 출력되는데, 이는 메일의 제목을 입력하라는 뜻
- 제목을 입력하면 'To: user1'이 자동으로 출력되고 그 다음 행으로 이동하여 기다리는데 이는 메일의 본문을 입력하라는 뜻
- 본문을 다 입력하고 Ctrl +d를 누르면 메일을 전송할 것인지 확인
- yes를 입력하면 메일이 전송
- 같은 메일을 여러 사용자에게 전송하려면 mailx 명령 다음에 수신자를 차례로 나열하면 됨

```
s-nail user1 user3 user4
```

03 메일 송수신

■ 메일 보내기

- 메일 내용을 파일로 이미 작성해 놓았으면 표준 입력 방향 전환을 이용
- 다음 예와 같이 하면 mail.txt 파일의 내용이 user1 사용자에게 전송됨

```
s-nail user1 < mail.txt
```

- 이때 메일 제목을 별도로 지정하고 싶으면 -s 옵션을 사용

```
s-nail -s test mail user1 < mail.txt
```

03 메일 송수신

■ 메일 보내기

- 메일을 작성하는 도중 취소하고 싶을 때는 Ctrl +c를 두 번 누름
- 취소된 메시지는 홈 디렉터리에 dead.letter라는 이름으로 저장되며, 나중에 다시 사용할 수 있음

```
[user2@localhost ~]$ s-nail user1
Subject: test mail2
To: user1
test
^^c
s-nail:
s-nail: (Interrupt -- one more to kill letter)
s-nail: There are new messages in the error message ring (denoted by ERROR),
s-nail:   which can be managed with the `errors' command
ERROR# /home/user2/dead.letter 3/42
```

03 메일 송수신

■ 메일 읽기

- 도착한 메일은 자동으로 기본 메일박스에 저장
- 로키 리눅스에서는 /var/spool/mail 디렉터리 아래에 사용자명으로 된 파일을 만들고 메일을 저장
- user1 사용자의 경우 /var/spool/mail/user1 파일에 새로 도착한 메일이 저장되는 것
- 도착한 메일이 있을 경우, 프롬프트에서 작업하면서 Enter 를 입력하면 메일이 있음을 알리는 메시지가 출력됨. 도착한 메일은 s-nail을 사용하여 읽을 수 있음

```
[user1@localhost ~]$  
/var/spool/mail/user1에 새로운 메일이 있습니다
```

- 메일이 없을 때 s-nail 명령을 입력하면 다음과 같이 메일이 없다는 메시지를 출력

```
[user2@localhost ~]$ s-nail  
s-nail: No mail for user2 at /var/spool/mail/user2
```

03 메일 송수신

■ 메일 읽기

표 12-2 s-nail의 내부 명령

내부 명령	의미
메일 번호	해당 번호의 메일을 읽는다.
mail [수신자 계정]	메일을 작성한다.
d [메일 번호]	메일을 삭제한다. 번호를 지정하면 해당 메일이 지워지고, 번호를 지정하지 않으면 마지막에 읽은 메일이 지워진다.
undelete [메일 번호]	지워진 메일을 복원한다.
Enter	다음 메일을 읽는다.
h	메일의 헤더 목록을 출력한다.
r	메일에 답장을 보낸다.
save [파일명]	메일을 지정한 파일명으로 저장한다.
q	변동 사항을 메일박스에 반영하고 종료한다.
x	변동 사항을 반영하지 않고 종료한다.
?	내부 명령의 목록을 보여준다.

03 메일 송수신

■ 메일 읽기

- user1이 받은 메일을 읽기 위한 기능을 살펴보자. 메일을 읽기 위해 s-nail 명령을 입력하면 다음과 같이 출력됨

```
[user1@localhost ~]$ s-nail
s-nail version v14.9.22. Type '?' for help
/var/spool/mail/user1: 1 message 1 new
▶N 1 user2@localhost.loca 2023-10-02 20:35 19/741 "Hello"
&
```

- 출력 내용의 첫 번째 행에는 s-nail의 버전과 도움말이 필요할 경우 '?'를 입력하라는 메시지가 출력
- 다음 행을 보면 한 개의 신규 메시지가 있음을 알 수 있음
- 세 번째 행의 메일 목록에서 각 항목의 의미는 [표 12-3]과 같음

03 메일 송수신

■ 메일 읽기

표 12-3 메일 목록의 출력 형식

항목	의미
▶	현재 메일 표시
N	메일의 상태 표시 • N: 새로 도착한 메일 • U: 이전에 수신했지만 아직 읽지 않은 메일 • R: 새로 수신하여 읽은 메일 • O: 이전에 읽은 메일로, 저장된 메일
1	메일 번호
user2@localhost.loca	메일을 보낸 사용자 계정
2023-10-02 20:35	메일이 전송된 날짜와 시간
19/741	헤더를 포함한 메일의 행 수(19행)와 문자 수(741자)
Hello	메일 제목

03 메일 송수신

■ 메일 읽기

- &는 s-nail 명령 내부에서 사용하는 프롬프트로, 여기서 사용할 수 있는 명령이 [표 12-2]의 s-nail 내부 명령. 메일을 읽으려면 Enter 를 누르거나 메일 번호 1을 입력하면 됨

```
& 1
[-- Message 1 -- 19 lines, 741 bytes --]:
From: user2@localhost.localdomain
Message-Id: <202310021135.392BZ7qi038676@localhost.localdomain>
Date: Mon, 02 Oct 2023 20:35:07 +0900
To: user1@localhost.localdomain
Subject: Hello

mail test
bye

&
```


03 메일 송수신

■ 답장 보내기

- s-nail에서 메일을 읽은 다음 답장을 보내는 내부 명령은 r
- 이전에 읽은 메일에 대한 답장을 보내려면 r 명령과 함께 해당 메일 번호를 지정해야 함
- 지금 읽은 메일에 대한 답장을 보내기 위해 r 명령을 입력하면 주소와 제목이 자동으로 나타남
- 답장 내용을 입력하고 Ctrl +d를 누르면 답장이 발송됨

03 메일 송수신

■ 답장 보내기

```
& r
To: user2@localhost.localdomain
Subject: Re: Hello
Cc: user1@localhost.localdomain

user2@localhost.localdomain wrote:

> mail test
> bye
mail test ok
good!
^D
-----
(Preliminary) Envelope contains:
To: user2@localhost.localdomain
Subject: Re: Hello
Cc: user1@localhost.localdomain
In-Reply-To: <202310021135.392BZ7qi038676@localhost.localdomain>
Send this message [yes/no, empty: recompose]? yes
&
```

03 메일 송수신

■ 메일 삭제하고 복구하기

- 읽은 메일은 삭제할 수 있으며 잘못 삭제한 메일을 복구할 수도 있음
- 읽은 메일을 보관하지 않고 삭제하려면 d 명령, 잘못 삭제한 메일을 복구하려면 undelete 명령 사용
- 1번 메일을 삭제했다가 다시 복구하는 예. 'd 1'을 입력하면 1번 메일이 삭제.
이때 'undelete 1'로 1번 메일을 복구하면 1번 메일은 이미 읽은 메일로 간주되어 메일의 상태가 R로 표시됨

```
& d 1
& undelete 1
& h
▶R 1 user2@localhost.local 2023-10-02 20:35 19/741 "Hello"
&
```

03 메일 송수신

■ 메일 종료하기

- 메일 작업을 마친 후 s-nail을 종료하려면 q나 x 명령을 사용
- q 명령으로 종료하면 읽은 메일과 읽지 않은 메일이 자동으로 저장
- 읽은 메일을 포함하여 모든 메일은 /var/ spool/mail/사용자명 파일에 저장됨

```
& q
New mail has arrived.
Held 1 message in /var/spool/mail/user1
/var/spool/mail/user1에 메일이 있습니다
[user1@localhost ~]$
```

Thank you!