



# Linux 명령줄

## Linux 기본 사항

# 학습 내용

## 강의 핵심 내용

학습 내용:

- 로그인 워크플로를 설명합니다.
- Linux 명령 구문을 설명합니다.
- 명령줄에서 기본 작업을 수행합니다.
- 스탠더드 입력/출력과 스탠더드 오류를 설명합니다.



# Linux 로그인 워크플로

# 로그인 프롬프트

- 사용자 이름과 암호를 입력합니다(시스템에 있는 파일 내용과 대조하여 확인됨).

```
password:
```

- 액세스가 허용되거나 거부됩니다.

```
CentOS Linux 7 (Core)  
Kernel 3.10.0-957.1.3.el7.x86_64 on an x86_64  
server00 login: _
```

# 로그인 워크플로

사용자 이름은 `/etc/passwd` 파일과 대조하여 확인됩니다.

```
login: c_salazar
```

암호는 `/etc/shadow` 파일과 대조하여 확인됩니다.

```
server00 login: userA
Password:
Last login: Thu Mar  7 21:28:52 on :0
[userA@server00 ~]$ pwd
/home/userA
```

`pwd` 명령은 현재 작업 디렉터리를 출력하는 유틸리티입니다.

# 로그인 워크플로

사용자 홈 디렉터리에 저장된 파일에서 사용자 프로필이 로드됩니다.

- `/home/username/.bashrc`
- `/home/username/.bash_history`
- `/home/username/.bash_profile`

```
server00 login: userA
Password:
Last login: Thu Mar  7 21:28:52 on :0
[userA@server00 ~]$ pwd
/home/userA
```

사용자의 현재 작업 디렉터리는 사용자 홈 디렉터리가 됩니다.



# 명령 프롬프트

# 명령 프롬프트 구조

```
login: c_salazar
```

```
Password:
```

```
[c_salazar@hostname ~]$ pwd
```

```
/home/c_salazar
```

```
[username@hostname ~]$
```



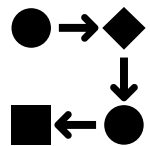
# 명령 구문의 예제

Linux 명령의 예는 다음과 같습니다.

man



man 명령은  
man 페이지 표시



명령: Linux에서  
수행하려는 작업



명령: 운전

-i

-i 옵션은 대소문자를  
구분하지 않는 검색 실행

옵션: 명령 수정

옵션: 헤드라이트 켜기

whoami

whoami 인수는 검색 대상

인수: 명령이 작동하는 대상

인수: 자동차



# 유용한 명령

# whoami 명령

```
[c_salazar@hostname ~]$ whoami
```

```
c_salazar
```

```
[c_salazar@hostname ~]$
```

# id 명령

```
[c_salazar@hostname ~]$ id
```

```
[userA@server00 ~]$ id  
uid=1003(userA) gid=1003(userA) groups=1003(userA) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[userA@server00 ~]$
```

id 명령은 실존하는 유효한 사용자 및 그룹 ID를 출력하는 데 사용됩니다.

# hostname 명령

```
[c_salazar@hostname ~]$ hostname
```

```
[userA@server00 ~]$ hostname  
server00  
[userA@server00 ~]$ █
```

hostname 명령은 시스템의 현재 호스트, 도메인, 노드 이름을 설정하거나 표시하는 데 사용됩니다.

# uptime 명령

```
[c_salazar@hostname ~]$ uptime
```

```
[userA@server00 ~]$ uptime  
21:39:05 up 11 min, 2 users, load average: 0.00, 0.12, 0.15  
[userA@server00 ~]$
```

uptime 명령은 시스템이 마지막으로 부팅된 이후 가동된 시간을 나타냅니다.

# date 명령

```
[c_salazar@hostname ~]$ date
```

```
[userA@server00 ~]$ date  
Thu Mar  7 21:37:34 GMT 2019  
[userA@server00 ~]$ █
```

date 명령은 현재 날짜와 시간을 제공합니다.

# cal 명령

```
[c_salazar@hostname ~]$ cal
```

```
[userA@server00 ~]$ cal
      March 2019
Su Mo Tu We Th Fr Sa
                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
[userA@server00 ~]$
```

cal 명령은 간단한 달력을 표시하는 데 사용됩니다.  
인수를 지정하지 않으면 현재 월이 표시됩니다.



# clear 명령

```
[c_salazar@hostname ~]$ clear
```

```
[c_salazar@hostname ~]$
```

clear 명령은 터미널 화면을 지우는 데 사용되는 시스템 명령입니다.

# echo 명령

```
[c_salazar@hostname ~]$ echo 'Hi!'
```

```
[c_salazar@hostname ~]$ Hi!
```

echo 명령은 터미널 화면에 따옴표 안의 문자열(*Hi!*)을 표시합니다.

```
[root@server00 ~]# echo 'Hello World!'  
Hello World!  
[root@server00 ~]#
```

# history 명령

```
[c_salazar@hostname ~]$ history
```

Bash는 해당 사용자의 홈 디렉터리에 있는 파일에 각 사용자의 명령 기록을 보관합니다. `history` 명령은 기록 파일을 확인합니다.

- 기록은 `/home/username/.bash_history`에 저장됩니다.

```
142 whoami
143 hostname
144 clear
145 man date
146 man cal
147 man cat
148 cat -n /proc/partitions
149 history
[userA@server00 ~]$ !143
hostname
server00
[userA@server00 ~]$
```

! 및 명령 번호는 기록 파일에서 특정 명령을 반복

현재 사용자의 기록 파일 표시

# touch 명령

```
[c_salazar@hostname ~]$ touch file_example_1
```

```
[userA@server00 Documents]$ ls
[userA@server00 Documents]$ touch file1.txt
[userA@server00 Documents]$ ls
file1.txt
[userA@server00 Documents]$ █
```

touch 명령은 각 파일의 액세스 시간과 수정 시간을 현재 시간으로 업데이트하는 데 사용됩니다.

# cat 명령

```
[c_salazar@hostname ~]$ cat
```

```
[root@server00 etc]# cat hosts.allow
#
# hosts.allow This file contains access rules which are used to
# allow or deny connections to network services that
# either use the tcp_wrappers library or that have been
# started through a tcp_wrappers-enabled xinetd.
#
# See 'man 5 hosts_options' and 'man 5 hosts_access'
# for information on rule syntax.
# See 'man tcpd' for information on tcp_wrappers
#
[root@server00 etc]# █
```

cat(concatenate) 명령은 파일에서 데이터를 읽고,  
터미널 창에 그 내용을 출력합니다.



# 추가 명령

# 스탠더드 입력 - stdin 명령

스탠더드 입력(stdin)

스탠더드 입력은 일반적으로 입력 수신 장치입니다(예: 키보드 또는 스캐너).



스탠더드 입력: 0

0은 스탠더드 입력에 사용하는 값입니다.

<는 myfirstscript 콘텐츠를 스탠더드로 리디렉트합니다.

```
[c_salazar@hostname ~]$ cat 0<myfirstscript
```

cat에 myfirstscript로부터 입력 정보를 가져오라고 지시합니다.

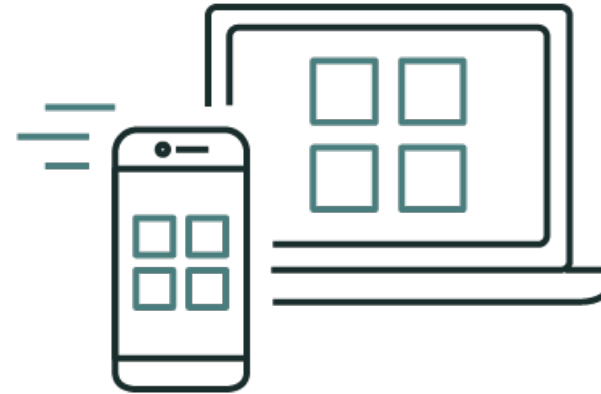
# 스탠더드 출력 - stdout 명령

스탠더드 출력(stdout)

스탠더드 출력은 일반적으로 출력이 전달되는 장치입니다(예: 디스플레이 모니터 또는 모바일 기기 화면).

스탠더드 출력: 1

1은 스탠더드 출력에 사용되는 값입니다.



```
[c_salazar@hostname ~]$ ls -l 1>folder.txt
```

`ls -l`은 현재 폴더의 내용을 나열합니다. 콘솔에 표시하는 대신, `ls`의 출력을 `folder.txt` 파일로 보냅니다.



# 스탠더드 오류 - stderr 명령

스탠더드 오류(stderr)

스탠더드 오류는 프로세스가 오류 메시지를 작성하는 위치입니다.



스탠더드 오류: 2

2는 스탠더드 오류에 사용되는 값입니다.

```
[c_salazar@hostname ~]$ find / -name "*" -print 2> / dev/null
```

참고: find 명령이 생성하는 모든 오류를 삭제하여 CLI를 깔끔하게 유지할 수 있습니다.

# Bash 탭 완성

```
[userA@server00 Documents]$ cat file  
file1 file2 file3 file4  
[userA@server00 Documents]$ cat file
```

중단한 부분에서 **탭**을  
두 번 눌러 원하는 특정  
파일을 선택합니다.

탭 키를 두 번 누르면 일치하는 모든 옵션이 표시됩니다.

# 모범 실무: 기록과 탭 완성



이 두 가지 기능을 사용하는 습관을 들이십시오.

# 확인 질문

Bash Shell 명령 구문을 구성하는 세 가지 요소는 무엇입니까?

Bash 기록에서 명령을 어떻게 실행합니까?

명령 문자열을 고유하게 만드는 데 충분한 문자를 입력하지 않은 경우, 명령에서 탭 완성을 시도하면 어떻게 됩니까?

# 요점



- 다음은 Linux 로그인 워크플로를 구성하는 3가지 주요 단계입니다.
  - 사용자에게 사용자 이름과 암호로 **인증**하라는 메시지가 표시됩니다.
  - 사용자 **프로필** 파일에 있는 사용자 세션 설정이 로드됩니다.
  - 사용자 **홈 디렉터리**에 **명령 프롬프트**가 표시됩니다.
- 명령 프롬프트에서 **탭** 키를 사용하여 **명령을 자동으로 완성**할 수 있습니다.
- 몇 가지 유용한 명령으로는 **history**, **whoami**, **hostname** 등이 있습니다.

# 감사합니다.

© 2021 Amazon Web Services, Inc. 또는 자회사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 임대 또는 판매는 금지됩니다. 수정해야 할 사항, 피드백 또는 기타 질문이 있다면 <https://support.aws.amazon.com/#/contacts/aws-training> 에서 문의해 주십시오. 모든 상표는 해당 소유자의 자산입니다.

