

오픈소스프로그래밍

Dong-Geol Choi

Information and Communication Engineering

Hanbat National University



프로세스와 사용자 명령 익히기



목차

1. 프로세스의 개념과 종류
2. 프로세스 관리 명령
3. 포그라운드와 백그라운드 프로세스
4. 사용자 정보 보기



학습목표

- 프로세스의 개념을 이해한다.
- 프로세스 관련 유닉스 명령의 사용 방법을 익힌다.
- 포그라운드 처리와 백드라운드 처리의 차이를 이해한다.
- 사용자 정보를 보는 명령의 사용 방법을 익힌다.



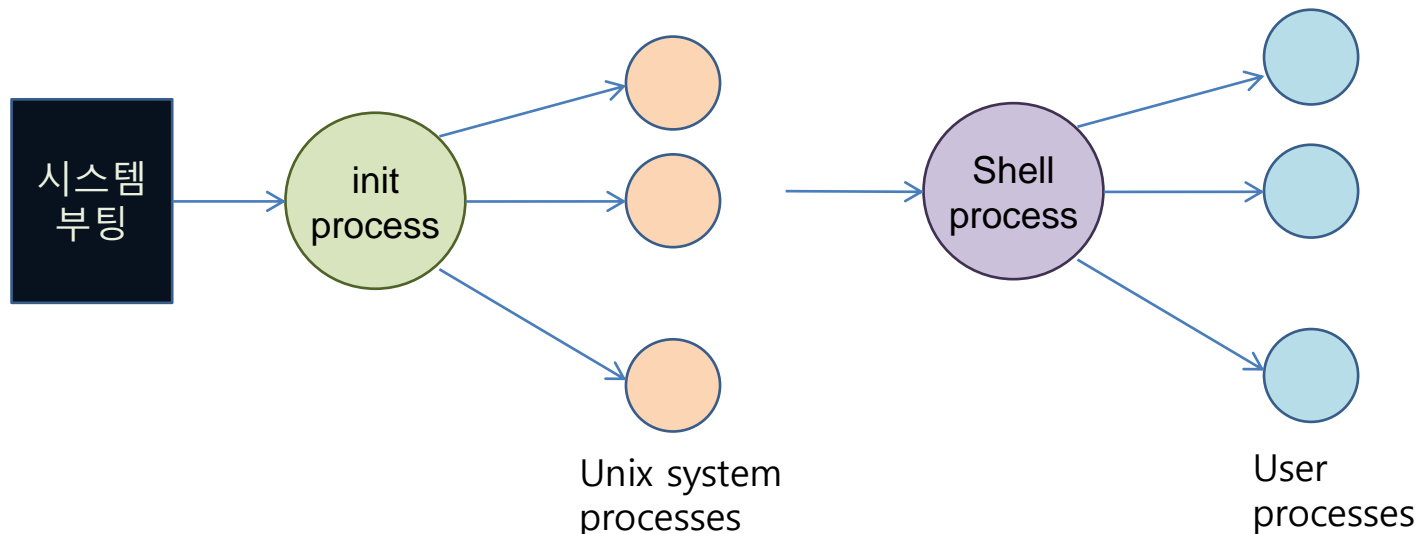
01. 프로세스의 개념과 종류

❖ 프로세스

▪ 실행중인 프로그램

▪ 종류

- 시스템 프로세스 : 유닉스 운영에 필요한 기능 수행
- 사용자 프로세스 : 사용자들이 실행시킨 프로세스



01. 프로세스의 개념과 종류

❖ 프로세스

프로세스	설명
데몬(daemon)	유닉스 커널에 의해 실행되는 프로세스로 특정 서비스 제공
부모(parent)	자식 프로세스를 만드는 프로세스
자식(child)	부모에 의해 생성된 프로세스 실행이 끝나면 부모 프로세스에 결과를 돌려주고 종료
고아(orphan)	실행도중에 부모 프로세스가 종료된 프로세스 고아프로세스는 1번 프로세스를 새로운 부모로 가진다
좀비(zombie)	부모프로세스가 종료처리를 하지 않은 프로세스 종료가 되었음에도 할당된 시스템 자원이 회수되지 않은 상태



02. 프로세스 관리 명령

❖ 프로세스 목록 보기

- ps
- pgrep

❖ 프로세스 종료 시키기

- kill
- pkill

❖ 포그라운드(전위)와 백그라운드(후위) 작업 제어

- fg
- bg
- jobs



02. 프로세스 관리 명령

❖ 프로세스 목록 보기

- ps

ps [옵션]

❖ process status

❖ 프로세스 정보를 출력

- PID, 터미널, CPU 시간, 명령어

❖ 옵션

- -e : 시스템에 있는 모든 프로세스 정보 출력
- -f : 프로세스에 대한 자세한 정보 출력
- -u UID : 특정 사용자에게 대한 모든 프로세스 출력



02. 프로세스 관리 명령

❖ 프로세스 목록 보기

▪ ps 사용 예

```
user1@Solaris11:~$ ps
  PID TTY          TIME CMD
 1429 pts/1        0:00 bash
 1525 pts/1        0:00 ps
user1@Solaris11:~$
```

```
user1@Solaris11:~$ ps -e
  PID TTY          TIME CMD
    0 ?              0:01 sched
    5 ?              0:05 zpool-rp
    6 ?              0:00 kmem_tas
    1 ?              0:00 init
```

```
user1@Solaris11:~$ ps -f
  UID    PID  PPID    C   STIME TTY          TIME CMD
  user1  1429  1428    0  13:19:06 pts/1        0:00 bash
  user1  1526  1429    0  21:16:50 pts/1        0:00 ps -f
user1@Solaris11:~$
```



02. 프로세스 관리 명령

❖ 프로세스 목록 보기

▪ ps 사용 예

```
user1@Solaris11:~$ ps -ef | more
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	0	0	0	05:55:53	?	0:01	sched
root	5	0	0	05:55:52	?	0:05	zpool-rpool
root	6	0	0	05:55:53	?	0:00	kmem_task
root	1	0	0	05:55:53	?	0:00	/usr/sbin/init

(중략)

```
user1@Solaris11:~$ ps -u user1
```

PID	TTY	TIME	CMD
1388	?	0:00	evolutio
1392	?	0:00	gsd-xran
1390	?	0:00	dconf-se
1114	vt/7	0:15	Xorg

(생략)



02. 프로세스 관리 명령

❖ 프로세스의 종류

▪ Ps 사용 예

표 8-1 ps -f의 출력 정보

항목	의미
UID	프로세스를 실행시킨 사용자 ID
PID	프로세스 번호
PPID	부모 프로세스 번호
C	프로세스의 우선순위
STIME	프로세스의 시작 시간을 '시:분:초' 형식으로 출력
TTY	프로세스가 실행된 단말기 종류와 번호
CMD	실행되고 있는 프로그램 이름(명령)



02. 프로세스 관리 명령

❖ 특정 프로세스 정보 검색하기 - pgrep

pgrep [옵션] 패턴

❖ 프로세스 이름으로 찾아 정보를 출력

- 솔라리스에만 있는 특별한 명령
- = ps [옵션] | grep 패턴

❖ 옵션

- -x : 패턴과 정확히 일치하는 프로세스 정보 출력
- -n : 패턴을 포함하고 있는 가장 최근의 프로세스 정보 출력
- -U uid : 특정 사용자의 모든 프로세스 출력
- -l : PID와 프로세스 이름 출력
- -t term : 특정 단말기와 관련된 프로세스 정보 출력

❖ 패턴

- 찾으려는 정보



02. 프로세스 관리 명령

❖ 특정 프로세스 정보 검색하기 - pgrep

```
user1@Solaris11:~$ ps -ef | grep bash
  user1  1444  1443    0 21:16:07 pts/1        0:00 bash
  user1  1446  1444    0 21:16:20 pts/1        0:00 grep bash
user1@Solaris11:~$
```

```
user1@Solaris11:~$ pgrep -l bash
1444 bash
user1@Solaris11:~$
```



02. 프로세스 관리 명령

❖ 프로세스 종료

- ps 명령으로 찾은 프로세스 중 불필요한 프로세스를 강제로 종료시킨다.
- 프로세스를 종료시키면 그 자식 프로세스들도 같이 종료된다.
- 프로세스를 종료시킬 때 PID나 프로세스 이름을 알아야 한다.

❖ 프로세스 종료 시키기

- kill
- pkill



02. 프로세스 관리 명령

❖ 프로세스 종료

- kill

```
kill [ 시그널 ] pid
```

❖ PID로 지정한 프로세스에게 시그널을 보냄

- 시그널을 받은 프로세스는 지정된 동작을 수행
- 시그널을 지정하지 않으면 프로세스를 종료시키는 15 번 시그널(SIGTERM)을 보냄
- 사용자의 프로세스만 종료시킬 수 있음
- 관리자(root)는 모든 프로세스를 종료시킬 수 있음



02. 프로세스 관리 명령

❖ 시그널

- 프로세스에게 보내는 신호
- 프로세스는 시그널을 수신하면 지정된 동작을 수행한다.
 - 예 : 신호 무시, 프로세스 종료, 일시 정지 등
- 종류

시그널 번호	시그널 이름	기능	기본 응답
1	SIGHUP	• 터미널 연결이 끊어진 경우에 발생	종료
2	SIGINT	• 보통 Ctrl-C에 의해 발생	종료
9	SIGKILL	• 프로세스를 강제종료 시킨다. • 이 시그널은 무시할 수 없다.	종료
15	SIGTERM	• 프로세스를 종료시킨다. • 이 시그널은 무시할 수도 있다. • kill 명령이 보내는 기본 시그널	종료



02. 프로세스 관리 명령

❖ 프로세스 종료

- **kill**

kill [시그널] 프로세스명

❖ 프로세스의 명령이름으로 프로세스를 찾아 지정한 시그널 보냄

- 솔라리스에서만 제공
- 사용자가 소유한 프로세스만 종료 가능

```
user1@Solaris11:~$ sleep 100&
[1] 1550
user1@Solaris11:~$ sleep 200&
```

```
[2] 1551
user1@Solaris11:~$ ps -f
      UID    PID  PPID   C   STIME TTY          TIME CMD
      user1  1444   1443   0  21:16:07 pts/1    0:00 bash
      user1  1552   1444   0  21:31:32 pts/1    0:00 ps -f
      user1  1550   1444   0  21:31:25 pts/1    0:00 sleep 100
      user1  1551   1444   0  21:31:30 pts/1    0:00 sleep 200
user1@Solaris11:~$ pkill sleep
[1]-  Terminated                  sleep 100
[2]+  Terminated                  sleep 200
user1@Solaris11:~$ ps -f
      UID    PID  PPID   C   STIME TTY          TIME CMD
      user1  1444   1443   0  21:16:07 pts/1    0:00 bash
      user1  1554   1444   0  21:31:56 pts/1    0:00 ps -f
user1@Solaris11:~$
```



[실습 5-1] 프로세스를 찾아서 종료시키기

1. 실습용 디렉토리 만들기

```
OpenSSH SSH client
user1@Solaris11:~$ mkdir -p Unix/ch8/Practice
user1@Solaris11:~$ cd Unix/ch8/Practice
/export/home/user1/Unix/ch8/Practice
user1@Solaris11:~/Unix/ch8/Practice$
```

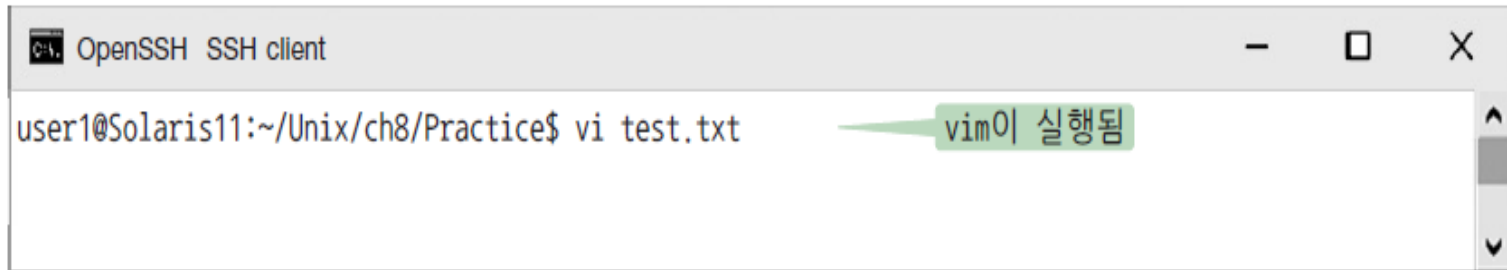
2. 파일 복사하기

```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ cp /etc/hosts test.txt
user1@Solaris11:~/Unix/ch8/Practice$
```



[실습 5-1] 프로세스를 찾아서 종료시키기

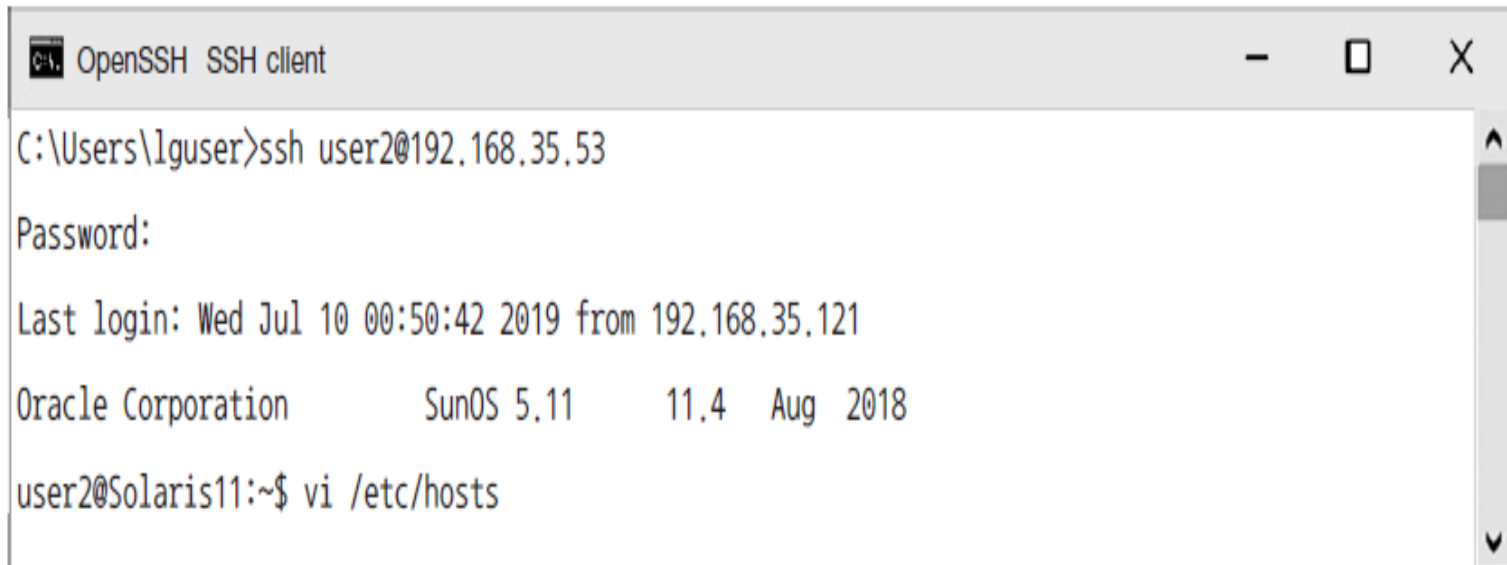
3. 프로세스 실행하기



A terminal window titled "OpenSSH SSH client" showing the command `vi test.txt` being executed. A green callout bubble points to the command with the text "vim이 실행됨" (vim is executed).

```
user1@Solaris11:~/Unix/ch8/Practice$ vi test.txt
```

4. 새로운 프로세스 실행하기



A terminal window titled "OpenSSH SSH client" showing the execution of an SSH command. The user logs in as user2 on a Solaris11 machine. The terminal displays the password prompt, the last login time, and the system version. Finally, the user executes the command `vi /etc/hosts`.

```
C:\Users\lguser>ssh user2@192.168.35.53
Password:
Last login: Wed Jul 10 00:50:42 2019 from 192.168.35.121
Oracle Corporation      SunOS 5.11      11.4   Aug  2018
user2@Solaris11:~$ vi /etc/hosts
```



[실습 5-1] 프로세스를 찾아서 종료시키기

5. 프로세스 찾기

```
OpenSSH SSH client
C:\Users\lguser>ssh user1@192.168.35.53
Password:
Last login: Wed Jul 10 00:53:42 2019 from 192.168.35.121
Oracle Corporation      SunOS 5.11      11.4      Aug 2018
user1@Solaris11:~$ ps -ef | grep vi
  user1  1486  1462   0 19:32:03 pts/1      0:00 vi test.txt
  user2  1511  1507   0 19:32:16 pts/2      0:00 vi /etc/hosts
user1@Solaris11:~$ pgrep vi
1486
1511
user1@Solaris11:~$ pgrep -u user1 vi
1486
user1@Solaris11:~$
```

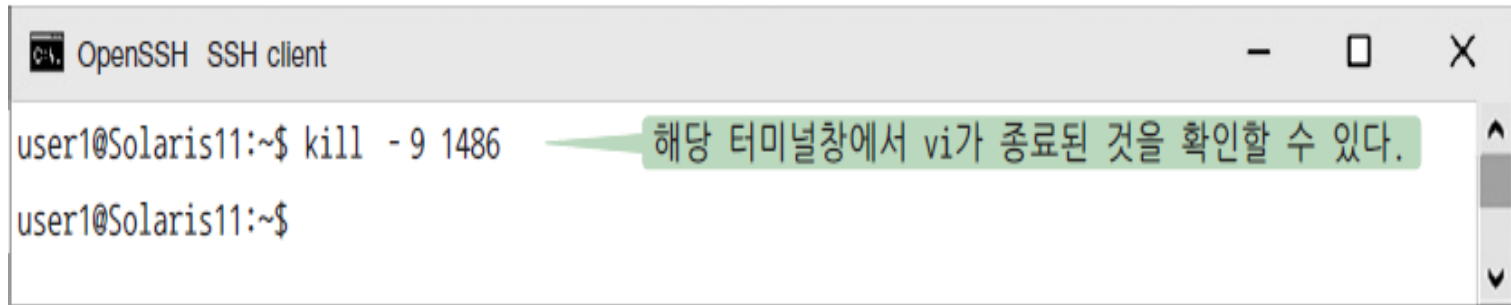
6. 프로세스 종료시키기

```
OpenSSH SSH client
user1@Solaris11:~$ kill -9 1511
bash: kill: (1511) - Not owner
user1@Solaris11:~$
```



[실습 5-1] 프로세스를 찾아서 종료시키기

7. 프로세스 종료시키기



A terminal window titled "OpenSSH SSH client" showing a user at the Solaris11 prompt. The user enters the command `kill -9 1486`. A green callout bubble points to the command with the text "해당 터미널창에서 vi가 종료된 것을 확인할 수 있다." (You can confirm that vi has ended in this terminal window). The prompt then changes to `user1@Solaris11:~$`.

```
user1@Solaris11:~$ kill -9 1486
user1@Solaris11:~$
```

8. 디렉토리 위치 변경하기



A terminal window titled "OpenSSH SSH client" showing a user at the Solaris11 prompt. The user enters the command `cd ..` from the directory `~/Unix/ch8/Practice`. The prompt then changes to `user1@Solaris11:~/Unix/ch8$`.

```
user1@Solaris11:~/Unix/ch8/Practice$ cd ..
user1@Solaris11:~/Unix/ch8$
```



02. 프로세스 관리 명령

❖ 프로세스 관리 도구

- top 명령

top [옵션]

- ❖ 주기적으로 현재 실행중인 프로세스에 대한 정보 출력
 - 솔라리스 11부터 기본 패키지로 탑재
- ❖ 자세한 요약 정보 출력
- ❖ 종료 : q



02. 프로세스 관리 명령

❖ 프로세스 관리 도구

▪ top 명령 사용 예

```
user1@Solaris11:~/Unix/ch8$ top
load averages:  0.06,  0.10,  0.08;                up 0+-7:-37:-52      21:57:03
142 processes: 137 sleeping, 4 running, 1 on cpu
CPU states: 91.6% idle,  3.2% user,  5.2% kernel,  0.0% stolen,  0.0% swap
Kernel: 547 ctxsw, 16 trap, 568 intr, 1800 syscall, 7 flt
Memory: 4096M phys mem, 1507M free mem, 4096M total swap, 4096M free swap
```

This terminal can only display 17 processes

PID	USERNAME	NLWP	PRI	NICE	SIZE	RES	STATE	TIME	CPU	COMMAND
1378	user1	10	59	0	420M	189M	run	0:31	1.66%	gnome-shell
746	root	21	59	0	178M	157M	sleep	0:58	1.06%	sstored
1200	user1	2	59	0	122M	57M	run	0:20	1.01%	Xorg
1481	user1	3	58	0	14M	2980K	run	0:08	0.33%	VBoxClient



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ 작업(job)

- 셸이 프로세스를 관리하는 단위
- 셸은 job을 시작시키고 제어 (Job ID 부여)
- 유닉스의 프로세스는 포그라운드 작업과 백그라운드 작업으로 구분가능

❖ 포그라운드 처리

- 사용자가 입력한 명령이 실행되어 결과가 출력될 때까지 프롬프트가 출력되지 않아서 다른 명령을 입력할 수 없는 처리 방법
- 보통의 처리방법

```
user1@Solaris11:~/Unix/ch8$ find / -name passwd ➡ 포그라운드 처리
```

```
➡ find 명령이 끝날 때까지 기다려야 함
```

❖ 백그라운드 처리

- 명령의 입력이 끝나면 결과에 상관없이 바로 다음 프롬프트가 출력되어 다른 명령을 입력할 수 있는 처리 방법
- 명령의 끝에 & 붙임

```
user1@Solaris11:~/Unix/ch8$ find / -name test.txt & ➡ 백그라운드 처리
```

```
[1] 1610
```

```
user1@Solaris11:~/Unix/ch8$ ➡ 프롬프트가 바로 나와 다른 명령을 실행시킬 수 있음
```



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ 작업제어 명령 - jobs

jobs [옵션]

❖ 작업 번호를 지정하지 않으면 백그라운드 작업을 모두 출력

❖ 특정 작업 번호를 지정하면 해당 작업의 정보만 출력

❖ 작업 번호

- %번호 : 해당 번호의 작업 정보를 출력
- %+ 또는 %% : 작업순서가 +인 작업 정보를 출력
- %- : 작업 순서가 -인 작업 정보를 출력

❖ 사용 예

```
user1@Solaris11:~/Unix/ch8$ sleep 100&
[1] 1652
user1@Solaris11:~/Unix/ch8$ sleep 200&
[2] 1653
user1@Solaris11:~/Unix/ch8$ jobs
[1]-  Running                  sleep 100 &
[2]+  Running                  sleep 200 &
```



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ jobs 명령 출력 항목

표 8-2 jobs 명령의 출력 항목

항목	출력 예	의미
작업 번호	[1]	작업 번호로 백그라운드로 실행시킬 때마다 순차적으로 증가한다([1] [2], [3]..).
작업 순서	+	작업 순서를 표시한다. <ul style="list-style-type: none">• + : 가장 최근에 접근한 작업• - : + 작업보다 바로 전에 접근한 작업• 공백 : 그 외의 작업
상태	Running	작업의 상태를 표시한다. <ul style="list-style-type: none">• 실행 중(Running) : 현재 실행 중이다.• 완료됨(Done) : 작업이 정상적으로 종료되었다.• 종료됨(Terminated) : 작업이 비정상적으로 종료되었다.• 정지(Stopped) : 작업이 잠시 중단되었다.
명령	sleep 100 &	백그라운드로 실행 중인 명령



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ 작업 전환 및 종료 명령

표 8-3 작업 전환 명령

명령	의미
<code>Ctrl+Z</code>	포그라운드 작업을 중지시킨다 (종료시키는 것이 아니라, 잠시 중단하는 것임).
<code>bg [%작업 번호]</code>	작업 번호가 지시하는 작업을 백그라운드 작업으로 전환시킨다.
<code>fg [%작업 번호]</code>	작업 번호가 지시하는 작업을 포그라운드 작업으로 전환시킨다.



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ 작업 전환 사용 예

```
user1@Solaris11:~/Unix/ch8$ jobs
user1@Solaris11:~/Unix/ch8$ sleep 150      ➡ 포그라운드 작업
^Z                                           ➡ Ctrl+Z로 작업을 정지시킴
[1]+  Stopped                  sleep 150
user1@Solaris11:~/Unix/ch8$ bg              ➡ 백그라운드 작업으로 전환
[1]+ sleep 150 &
user1@Solaris11:~/Unix/ch8$ jobs
[1]+  Running                  sleep 150 &
user1@Solaris11:~/Unix/ch8$
```



03. 프로세스 관리 명령

❖ 포그라운드와 백그라운드 작업의 이해

❖ 작업제어 명령 - nohup

nohup 명령&

- ❖ 로그아웃을 한 뒤에도 백그라운드 작업을 계속 실행
 - 원래는 로그아웃하면 실행중이던 모든 작업이 종료됨
- ❖ 명령 실행결과와 오류메시지 : nohup.out 파일에 저장
- ❖ 사용예

```
user1@Solaris11:~/Unix/ch8$ nohup find / -name passwd &
```

```
[1] 1661
```

```
Sending output to nohup.out ➡ 메시지 출력
```

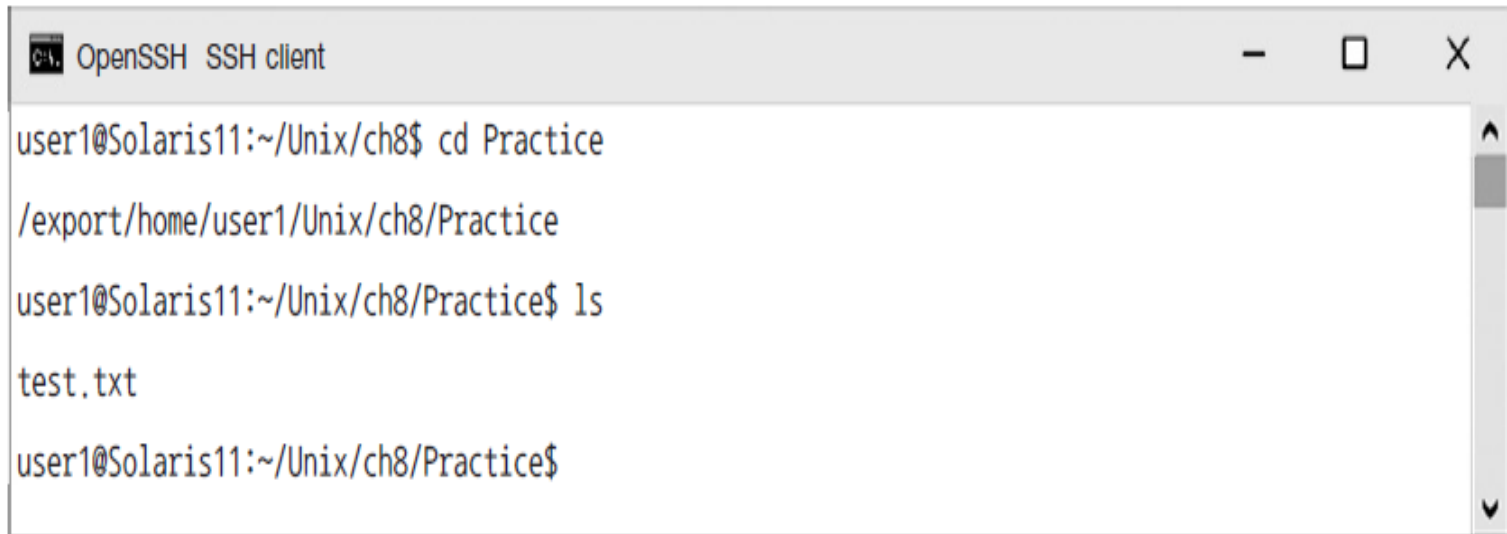
```
user1@Solaris11:~/Unix/ch8$
```

- ./nohup.out 파일에 결과 저장.
- 다음 로그인할 때 nohup.out 파일 확인



[실습 5-2] 작업 관리하기

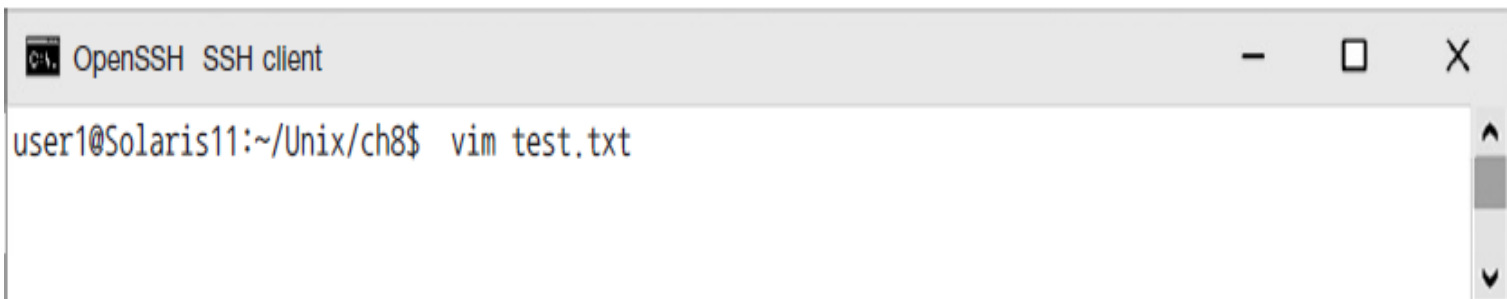
1. 실습용 디렉토리 만들기



```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8$ cd Practice
/export/home/user1/Unix/ch8/Practice
user1@Solaris11:~/Unix/ch8/Practice$ ls
test.txt
user1@Solaris11:~/Unix/ch8/Practice$
```

A terminal window titled "OpenSSH SSH client" showing a user navigating to a directory and listing its contents. The user is at the prompt "user1@Solaris11:~/Unix/ch8\$". They enter "cd Practice", and the prompt changes to "user1@Solaris11:~/Unix/ch8/Practice\$". They then enter "ls", and the output "test.txt" is displayed. The prompt returns to "user1@Solaris11:~/Unix/ch8/Practice\$".

2. 백그라운드 작업 만들기(1)

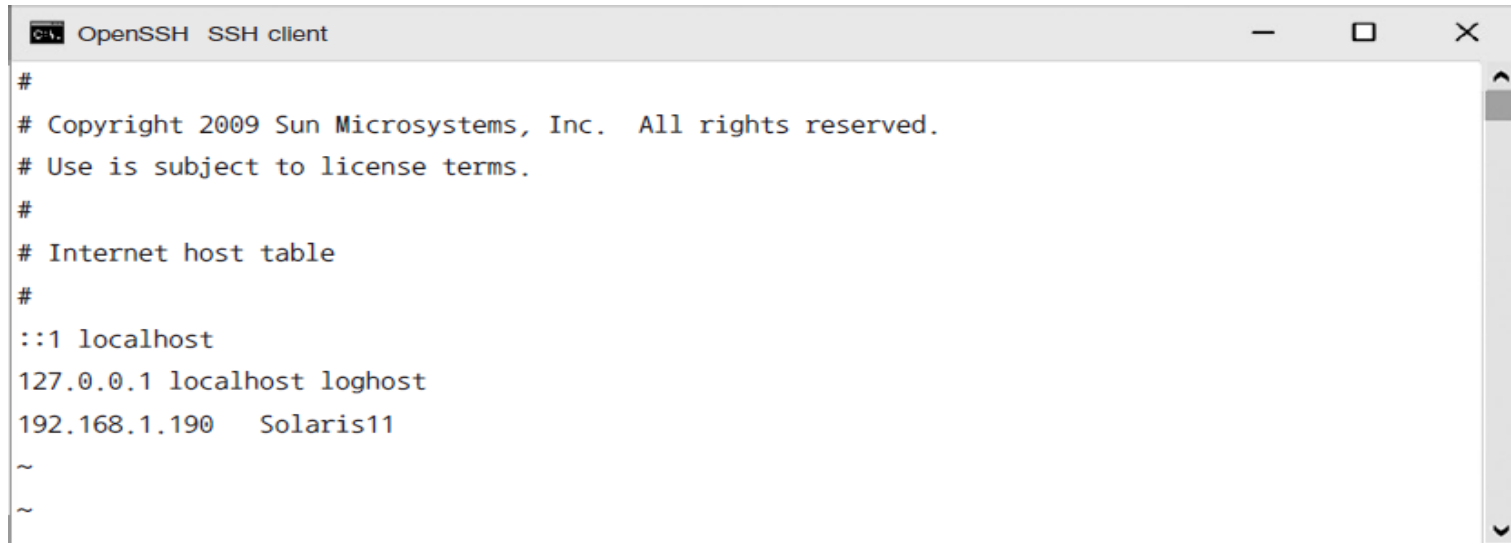


```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8$ vim test.txt
```

A terminal window titled "OpenSSH SSH client" showing a user starting a background job. The user is at the prompt "user1@Solaris11:~/Unix/ch8\$". They enter "vim test.txt".

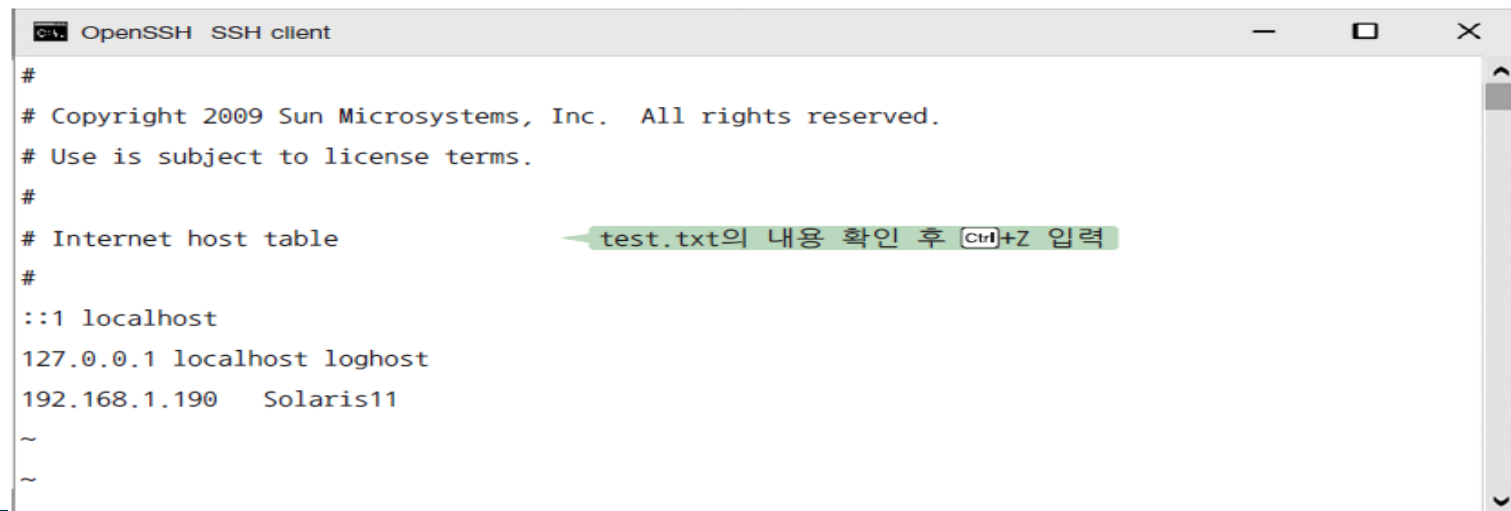


[실습 5-2] 작업 관리하기



```
OpenSSH SSH client
#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Internet host table
#
::1 localhost
127.0.0.1 localhost loghost
192.168.1.190 Solaris11
~
~
```

3. 백그라운드 작업 만들기(2)

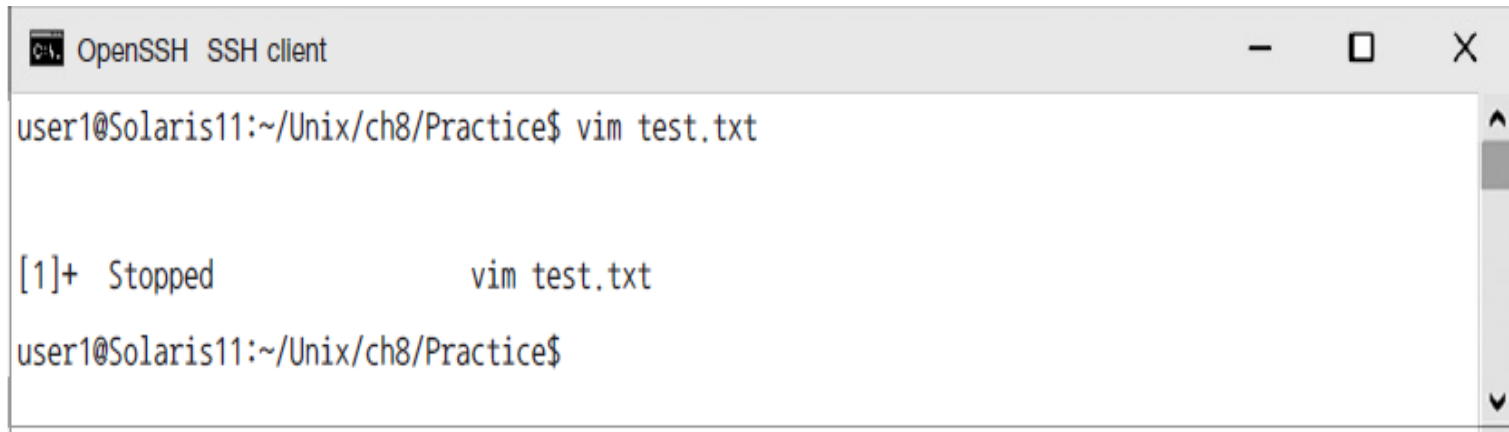


```
OpenSSH SSH client
#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Internet host table
#
::1 localhost
127.0.0.1 localhost loghost
192.168.1.190 Solaris11
~
~
```

test.txt의 내용 확인 후 **Ctrl+Z** 입력



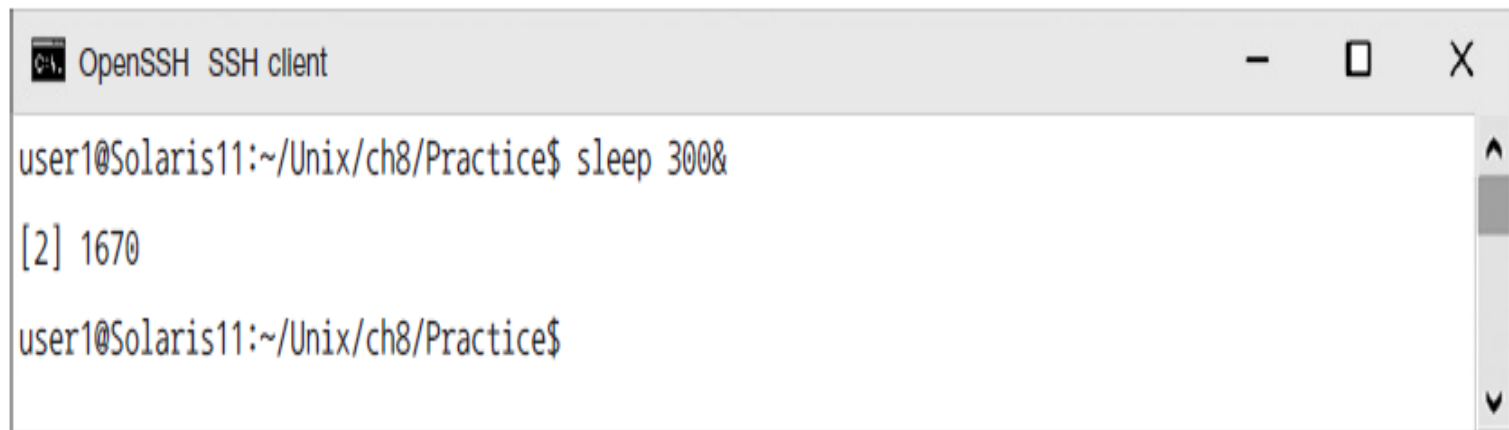
[실습 5-2] 작업 관리하기



```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ vim test.txt

[1]+  Stopped                  vim test.txt
user1@Solaris11:~/Unix/ch8/Practice$
```

4. 명령을 백그라운드로 실행시키기



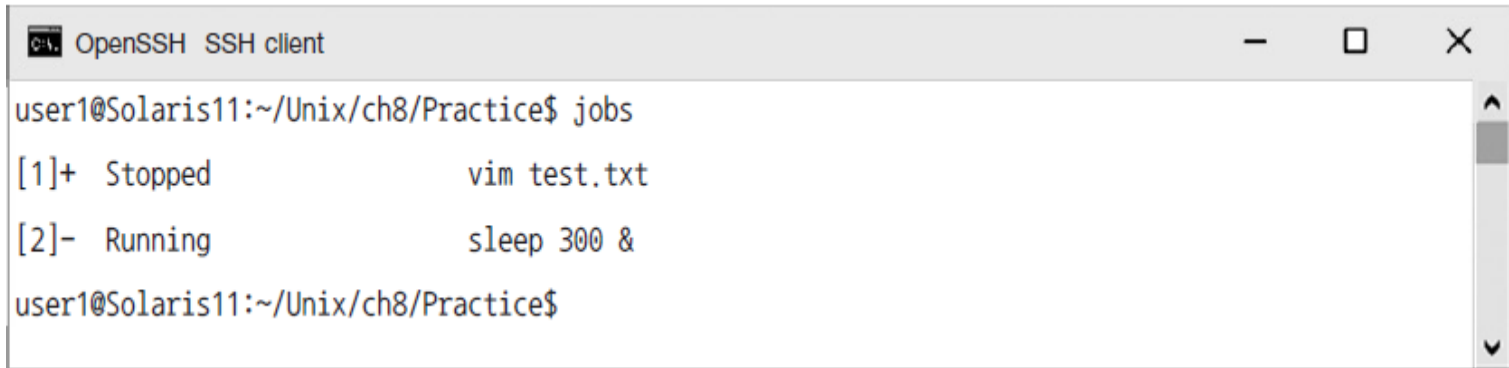
```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ sleep 300&

[2] 1670
user1@Solaris11:~/Unix/ch8/Practice$
```



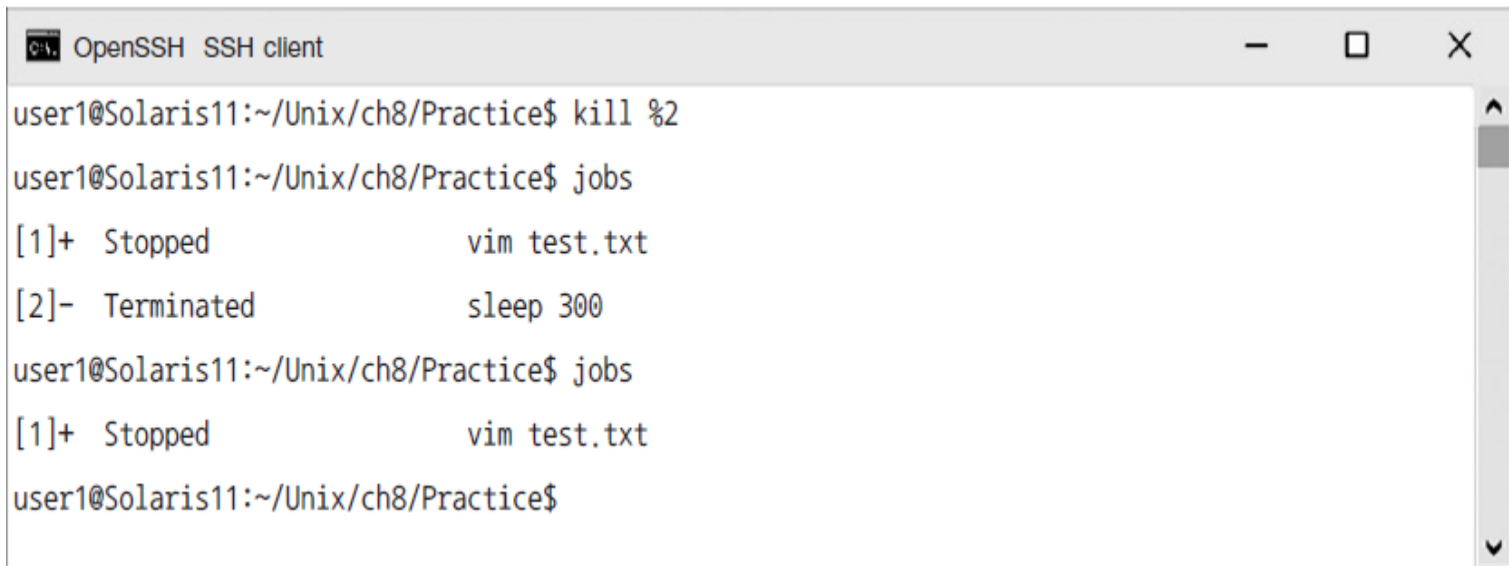
[실습 5-2] 작업 관리하기

5. 작업 목록 확인하기

A terminal window titled "OpenSSH SSH client" showing the output of the 'jobs' command. The prompt is 'user1@Solaris11:~/Unix/ch8/Practice\$'. The output shows two jobs: [1]+ Stopped vim test.txt and [2]- Running sleep 300 &. The prompt returns to 'user1@Solaris11:~/Unix/ch8/Practice\$' after the command.

```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ jobs
[1]+  Stopped          vim test.txt
[2]-  Running          sleep 300 &
user1@Solaris11:~/Unix/ch8/Practice$
```

6. 작업 종료시키기

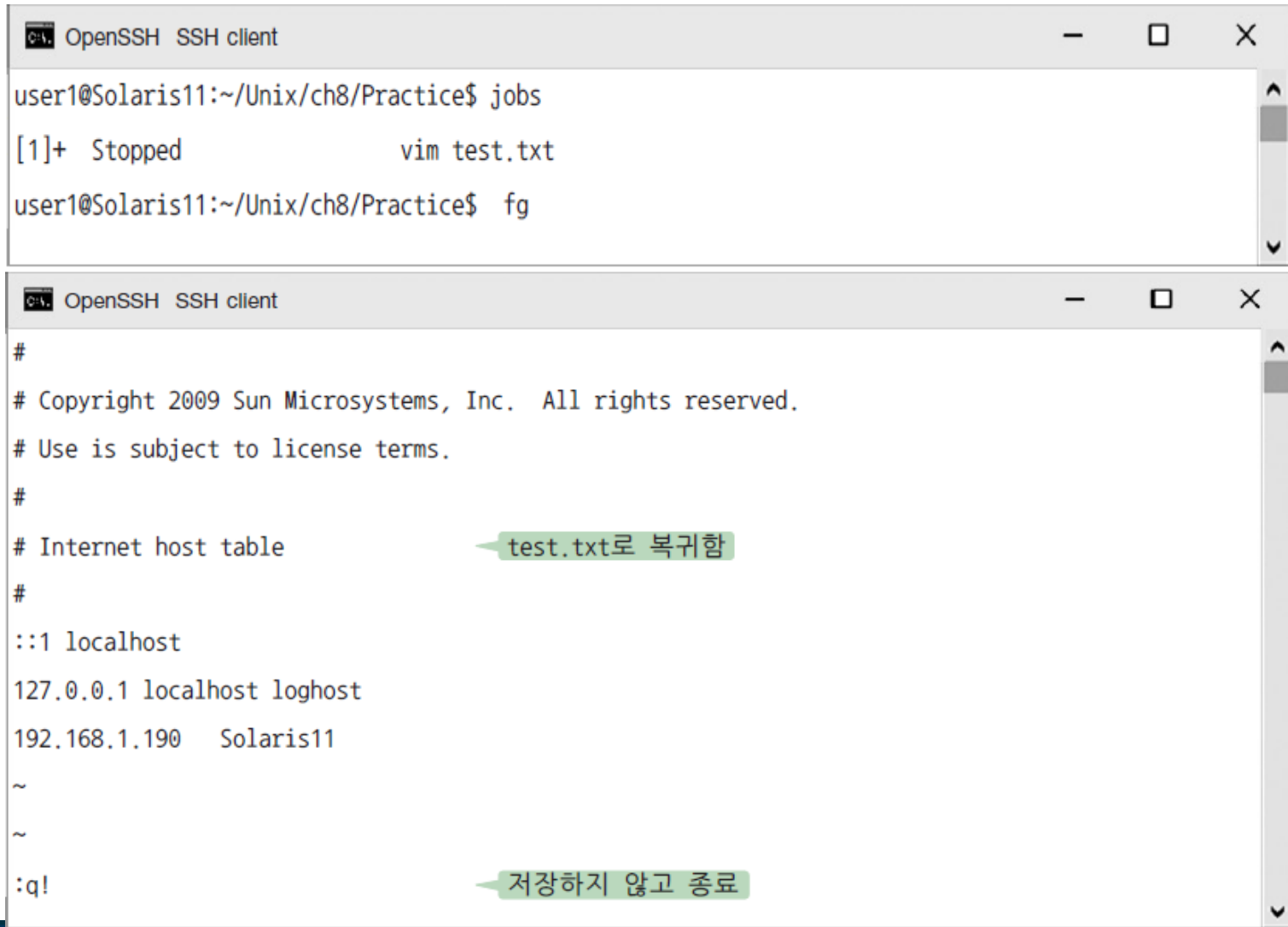
A terminal window titled "OpenSSH SSH client" showing the process of terminating a job. The prompt is 'user1@Solaris11:~/Unix/ch8/Practice\$'. The user enters 'kill %2'. The prompt returns to 'user1@Solaris11:~/Unix/ch8/Practice\$'. The user then enters 'jobs', and the output shows [2]- Terminated sleep 300. The user enters 'jobs' again, and the output shows [2] is no longer listed. The prompt returns to 'user1@Solaris11:~/Unix/ch8/Practice\$' after the final 'jobs' command.

```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ kill %2
user1@Solaris11:~/Unix/ch8/Practice$ jobs
[1]+  Stopped          vim test.txt
[2]-  Terminated      sleep 300
user1@Solaris11:~/Unix/ch8/Practice$ jobs
[1]+  Stopped          vim test.txt
user1@Solaris11:~/Unix/ch8/Practice$
```



[실습 5-2] 작업 관리하기

7. 작업 목록 확인하기



The image shows two terminal windows from an OpenSSH SSH client. The top window shows the execution of the 'jobs' command, which lists a stopped job '[1]+ Stopped vim test.txt', followed by the 'fg' command to bring it back to the foreground. The bottom window shows the output of the 'fg' command, which is the vim editor's startup screen. It includes copyright information and a host table. A green callout bubble points to 'test.txt' in the host table, saying 'test.txt로 복귀함' (Return to test.txt). Another green callout bubble points to the ':q!' command at the bottom, saying '저장하지 않고 종료' (Exit without saving).

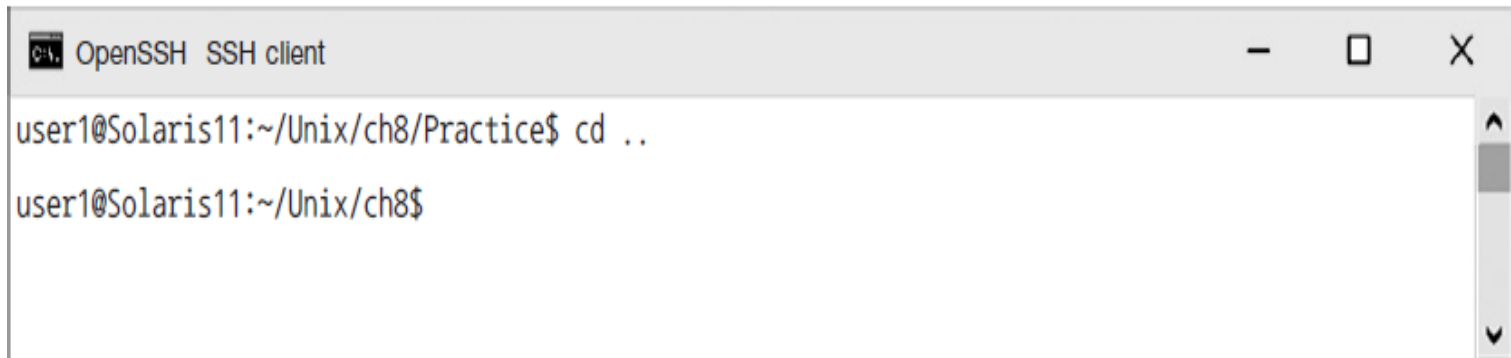
```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ jobs
[1]+  Stopped                  vim test.txt
user1@Solaris11:~/Unix/ch8/Practice$ fg

#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Internet host table      test.txt로 복귀함
#
::1 localhost
127.0.0.1 localhost loghost
192.168.1.190  Solaris11
~
~
:q!                          저장하지 않고 종료
```



[실습 5-2] 작업 관리하기

8. 디렉토리 위치 변경하기

A terminal window titled "OpenSSH SSH client" with standard window controls. It shows a user at a Solaris11 machine. The prompt is "user1@Solaris11:~/Unix/ch8/Practice\$". The user enters "cd .." and the prompt changes to "user1@Solaris11:~/Unix/ch8\$".

```
OpenSSH SSH client
user1@Solaris11:~/Unix/ch8/Practice$ cd ..
user1@Solaris11:~/Unix/ch8$
```



04. 사용자 정보 보기

❖ 로그인한 사용자 정보보기

- users
- who
- w

❖ 사용자 자신의 정보보기

- who am i
- whoami
- Id

❖ 사용자 전환하기

- su



04. 사용자 정보 보기

❖ 로그인한 사용자 정보보기

- `users` – 사용자명 출력하기

`users`

❖ 현재 시스템에 로그인 하고 있는 사용자명을 출력

❖ 사용 예

```
user1@Solaris11:~/Unix/ch8$ users  
user1 user1 user2  
user1@Solaris11:~/Unix/ch8$
```



04. 사용자 정보 보기

❖ 로그인한 사용자 정보보기

- **who** - 사용자 정보 출력하기

who [옵션]

❖ 시스템을 사용하고 있는 사용자의 정보를 출력

❖ 옵션

- **-q** : 사용자명만 출력한다.
- **-H** : 출력항목의 제목도 함께 출력한다.
- **-b** : 마지막으로 재부팅한 날짜와 시간을 출력한다.

❖ 사용 예

```
user1@Solaris11:~/Unix/ch8$ who
user1      pts/1        Jul 10 19:32    (192.168.35.121)
user2      pts/2        Jul 10 19:32    (192.168.35.121)
user1      pts/3        Jul 10 19:13    (192.168.35.121)
user1@Solaris11:~/Unix/ch8$
```

```
user1@Solaris11:~/Unix/ch8$ who -q
user1      user2        user1
# users=3
user1@Solaris11:~/Unix/ch8$
```

```
user1@Solaris11:~/Unix/ch8$ who -b
.          system boot  Jul 10 09:21
user1@Solaris11:~/Unix/ch8$
```



04. 사용자 정보 보기

❖ 로그인한 사용자 정보보기

- w - 사용자 작업 정보 출력하기

w [사용자명]

❖ 로그인한 사용자 정보와 현재 작업의 정보를 출력

❖ 사용 예

```
user1@Solaris11:~/Unix/ch8$ w
  1:32am  up  5 users,  load average: 0.04, 0.05, 0.06
User      tty          login@  idle   JCPU   PCPU   what
-----
user1     vt/7          12:30am 1:03    58     48    /usr/lib/tracker-miner-apps
user2     pts/2          12:53am  21          -bash
user1     pts/3          1:02am  30          -bash
user1@Solaris11:~/Unix/ch8$
```



04. 사용자 정보 보기

❖ 사용자 자신의 정보 보기

- whoami – 자신의 로그인 사용자명 출력하기

```
whoami
```

❖ 사용 예

```
user1@Solaris11:~/Unix/ch8$ whoami  
user1  
user1@Solaris11:~/Unix/ch8$
```



04. 사용자 정보 보기

❖ 사용자 전환하기

- su - 다른 사용자로 전환하기

```
su [ - ] [ 사용자명 ]
```

❖ 사용자 계정을 변경

- 아무 옵션 없이 사용하면 : root 계정으로 변경
- 사용자명 지정 : 지정한 사용자로 변경
- - : root 혹은 지정한 사용자의 초기화 파일이 실행되면서 작업 디렉토리로 사용자의 홈 디렉토리로 변경됨

```
user1@Solaris11:~/Unix/ch8$ su user2
Password:
user2@Solaris11:~/Unix/ch8$ whoami
user2
user2@Solaris11:~/Unix/ch8$ pwd
/export/home/user1/Unix/ch8
user2@Solaris11:~/Unix/ch8$ exit
exit
user1@Solaris11:~/Unix/ch8$ whoami
user1
user1@Solaris11:~/Unix/ch8$
```

```
user1@Solaris11:~/Unix/ch8$ su - user2
Password:
Oracle Corporation      SunOS 5.11      11.4      Aug 2018
user2@Solaris11:~$ pwd
/export/home/user2
user2@Solaris11:~$
```

```
user2@Solaris11:~$ su
Password:
root@Solaris11:~# whoami
root
```

