



프로세스 관리

Linux 기본 사항

학습 내용

강의 핵심 내용

학습 내용:

- Linux의 프로세스를 정의합니다.
- 프로세스 관리에 필요한 기본 명령을 설명합니다.
- 작업 스케줄링에 필요한 `at` 명령과 `cron` 명령을 비교합니다.

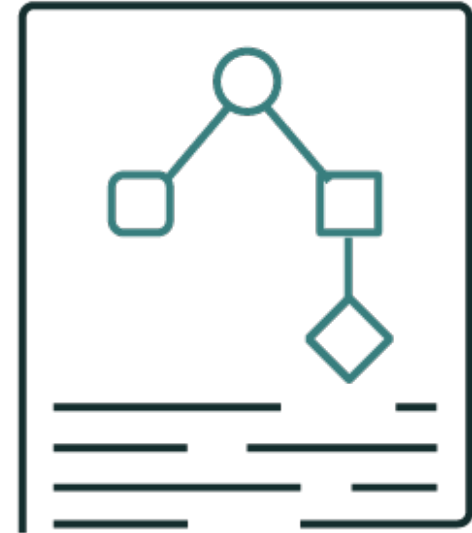




프로세스란?

프로그램

- 프로그램은 컴퓨터가 수행해야 할 액션을 알려주는 여러 가지 명령입니다.
- 프로그램은 일반적으로 다음의 두 가지 중 하나에 해당합니다.
 - 시스템 프로그램
 - 애플리케이션 프로그램



프로그램을 찾는 방법

```
]$ echo $PATH
```

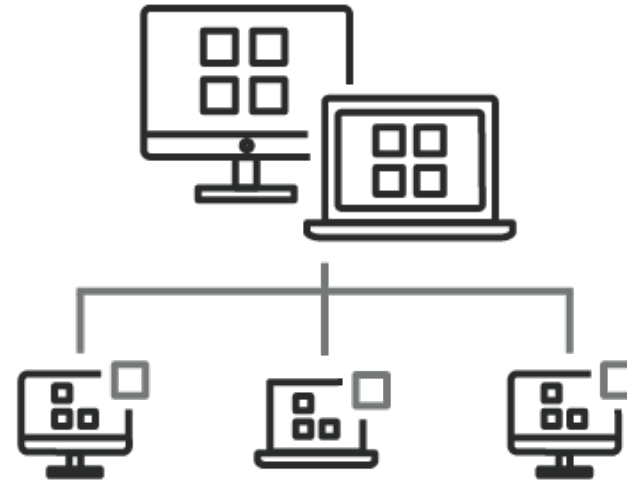
명령이 실행되면 시스템에서 다음을 수행합니다.

- 지정된 경로 찾기
- 프로그램의 \$PATH 변수 검색

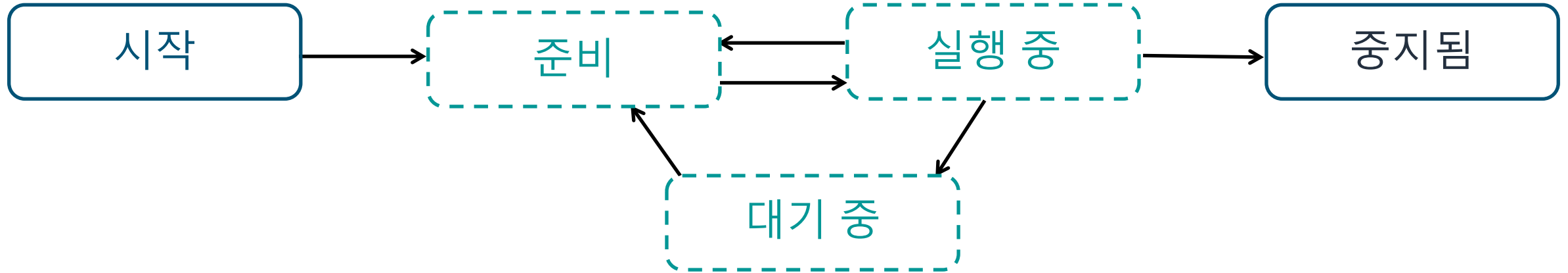
```
[ec2-user]$ echo $PATH  
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/ec2-user/.local/bin:/home/  
ec2-user/bin  
[ec2-user]$
```

프로세스란?

- 실행 중인 프로그램
- 프로세스 ID 번호(PID)로 식별
- 아래의 명령으로 확인 가능
 - ps
 - pstree



프로세스 상태



시작

프로세스가 만들어졌습니다.

준비

프로세스가 할당되기를 기다립니다.

실행 중

프로세스가 진행 중입니다.

중지됨

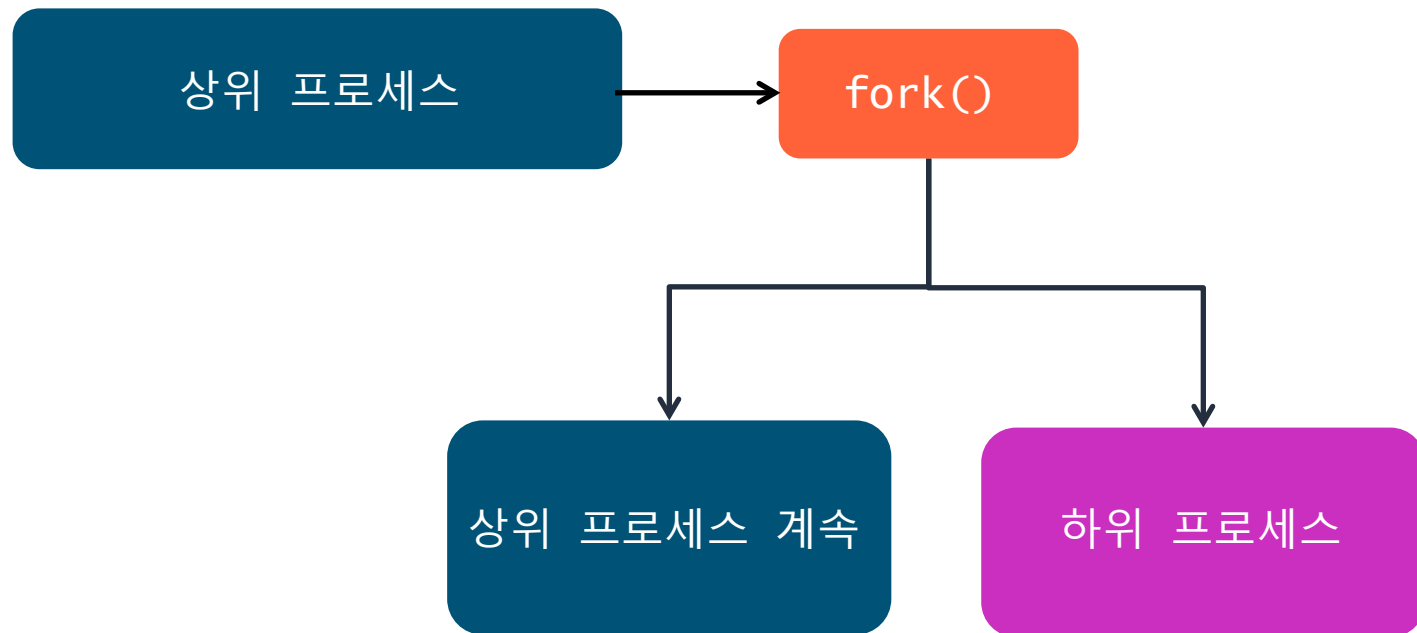
모든 작업이 종료되었습니다.

대기 중

프로세스가 이벤트를 기다리고 있습니다.

하위 프로세스란?

일부 서비스와 애플리케이션은 복잡하고, 다양한 기능을 제공하려면 둘 이상의 프로세스가 필요합니다. 이러한 서비스는 하위 프로세스를 만들어 냅니다.



프로세스 관리에 필요한 기본 명령

ps 명령

```
]$ ps [options]
```

- 운영 체제의 현재 프로세스 표시

```
[ec2-user]$ ps
```

PID	TTY	TIME	CMD
31532	pts/0	00:00:00	bash
31590	pts/0	00:00:00	ps

일반적인 ps 명령 옵션

옵션	설명
-e	도움말과 버전
-b	배치 모드 사용
-fp <number>	PID로 프로세스 나열

pidof 명령

```
]$ pidof [options] programName
```

- 실행 중인 프로그램의 프로세스 ID(PID)를 표시합니다.

```
]$ pidof sshd
```

- 이 예에서 `pidof sshd`는 `sshd`의 PID를 출력합니다.

```
[ec2-user]$ pidof sshd  
31531 31499 2431
```

ps tree 명령

```
]$ pstree [options] [pid, user]
```

명령이 실행되면 다음을 수행합니다.

- 현재 실행 중인 프로세스를 트리 형식으로 표시
- 상위([]로 표시) 프로세스와 하위({ }로 표시) 프로세스의 식별 지원

```
[ec2-user]$ pstree
systemd─acpid
        └─2*[agetty]
amazon-ssm-agent─ssm-agent-work─8*[{ssm-agent-work}]
                └─8*[{amazon-ssm-agent}]
anacron
atd
auditd─{auditd}
chronyd
crond
dbus-daemon
2*[dhclient]
gssproxy─5*[{gssproxy}]
irqbalance─{irqbalance}
lsmd
lvmetad
master─pickup
        └─qmgr
rngd
rpcbind
rsyslogd─2*[{rsyslogd}]
sshd─sshd─sshd─bash─pstree
systemd-journal
systemd-logind
systemd-udev
```

top 명령

```
]$ top [options]
```

top 명령은 시스템 성능과 활용도에 대한 실시간 요약 정보를 표시하고, 시스템에서 활성 상태인 프로세스와 스레드를 나열합니다.

```
top - 17:45:19 up 46 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 88 total, 1 running, 49 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 3977796 total, 3366568 free, 121276 used, 489952 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 3635080 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	125528	5424	4012	S	0.0	0.1	0:01.31	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.11	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0

일반적인 top 옵션

이러한 옵션을 사용하여 실행 중인 시스템의 다양한 측면을 확인하고 변경합니다.

옵션	설명
-h 및 -v	사용 방법과 버전 정보 표시
-b	배치 모드에서 top 시작

```
[ec2-user]$ top -hv  
procps-ng version 3.3.10
```

```
Usage:  
top -hv | -bCHiOss -d secs -n max -u|U user -p pid(s) -o field -w [cols]
```

```
[ec2-user]$ top -b
```

```
top - 17:46:31 up 47 min, 1 user, load average: 0.00, 0.00, 0.00  
Tasks: 88 total, 1 running, 49 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 3.2 sy, 0.0 ni, 96.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
KiB Mem : 3977796 total, 3366940 free, 120900 used, 489956 buff/cache  
KiB Swap: 0 total, 0 free, 0 used. 3635456 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND  
    1 root        20   0 125528   5424  4012 S   0.0   0.1   0:01.31 systemd  
    2 root        20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd  
    4 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/0:0H  
    6 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 mm_percpu_wq
```

top 태스크 상태

top 태스크(프로세스 또는 스레드)는 다음 상태 중 하나일 수 있습니다.

running

프로세스가 CPU에서 실행 중이거나 실행 대기열에 있습니다.

sleep

프로세스가 I/O 작업이 완료되기를 기다리고 있습니다.

stopped

작업 제어 신호로 프로세스가 중지되었거나 추적 중입니다.

zombie

상위 프로세스가 종료된 하위 프로세스입니다.

출력

```
top - 20:12:28 up 3:13, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 90 total, 1 running, 49 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.3 st
KiB Mem : 3977796 total, 3359996 free, 122140 used, 495660 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 3634520 avail Mem
```


top CPU 값

값	설명
us 값	사용자 공간에서 프로세스를 실행하는 데 소요된 시간
sy 값	커널 공간 프로세스를 실행하는 데 소요된 시간
id 값	유휴 시간
wa 값	I/O가 완료될 때까지 기다린 시간
hi 값	하드웨어 문제를 처리한 시간
si 값	소프트웨어 문제를 처리한 시간
st 값	다른 CPU 프로세스가 완료될 때까지 기다리며 소비한 시간

출력

```
top - 20:12:28 up 3:13, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 90 total, 1 running, 49 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.3 st
KiB Mem : 3977796 total, 3359996 free, 122140 used, 495660 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 3634520 avail Mem
```

top 메모리 사용 및 스와프

메모리

이 옵션은 시스템
메모리 사용량을 보여
줍니다.

스와프 공간

이 옵션은 하드
디스크의 랜덤 액세스
메모리(RAM)처럼
사용됩니다.

RAM 사용량

RAM을 거의 다 소진하면
사용 빈도가 낮은 리전이
스와프 공간에 덮어 쓰워져
검색될 때까지 대기합니다.

출력

```
top - 20:12:28 up 3:13, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 90 total, 1 running, 49 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.3 st
KiB Mem : 3977796 total, 3359996 free, 122140 used, 495660 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 3634520 avail Mem
```

kill 명령

```
]$ kill [options] processID
```

이 명령이 실행되면 다음이 수행됩니다.

- 일반적으로 프로세스가 자체 종료되지 않을 때 확실하게 프로세스를 종료합니다.
 - 상위 프로세스가 종료되면 보통 하위 프로세스도 종료됩니다.
- 자주 사용되는 kill 명령은 다음과 같습니다.
 - -9 SIGKILL – 천천히 종료하지 않고 즉시 프로세스 종료
 - -15 SIGTERM – 즉시 종료하지 않고 정상적으로 종료
 - -19 SIGSTOP – 프로세스를 일시 중지하고 명령줄을 사용할 수 있음

```
[ec2-user]$kill -9 32198
```

nice 및 renice 명령

```
]$ nice [options] [cmd]
```

- 이 명령은 지정된 우선순위로 새 프로세스를 시작합니다.
- 가장 높은 순위가 -20, 가장 낮은 순위가 19입니다.

```
]$ renice [options] [cmd]
```

- 이 명령은 이미 실행 중인 프로세스의 우선순위를 조정합니다.

```
[ec2-user]$ nice -5 bash  
[ec2-user]$ pidof bash  
32271 32145
```

데모: 다양한 도구를 이용한 프로세스 관리

시스템에서 실행 중인 코드의 각 인스턴스를 프로세스라고 합니다. `ps`, `pstree`, `top`를 사용하여 시스템의 프로세스를 살펴봅니다.



jobs 명령

```
]$ jobs
```

- jobs는 사용자가 시작하고 관리하는 프로세스로 작업 번호에 따라 식별됩니다.
 - jobs 명령은 다음 구문을 사용하여 작업 프로세스 번호를 나열합니다.
 - bg [job process number or name]: 백그라운드에서 실행
 - fg [job process number or name]: 포어그라운드에서 실행
 - 이러한 프로세스를 관리하려면 jobs 명령을 사용합니다.

at 및 cron 명령

]\$ at [options]

이 명령은 지정된 시간에 태스크를 한 번 실행합니다.

- 이 명령은 예컨대 '오늘만 오후 4시에 백업 스크립트 실행'하는 등의 일회성 태스크에 사용합니다.
- 예정된 작업을 표시하려면 `at -l` 옵션을 사용합니다.
- 예정된 작업을 삭제하려면 `atrm [number]` 명령을 사용합니다.

```
[root@server00 ~]# at 1600
at> ./demo-script.sh
at> <EOT>
job 2 at Wed Mar  6 16:00:00 2019
[root@server00 ~]# at -l
2      Wed Mar  6 16:00:00 2019 a root
[root@server00 ~]#
[root@server00 ~]# at -l
```

]\$ cron [option]

이 명령은 지정된 시간에 태스크를 정기적으로 실행합니다.

- 이 명령은 예컨대 '매주 오후 4시에 백업 스크립트 실행'하는 등의 반복적인 태스크에 사용합니다.
- 이 명령은 `crontab` 파일에서 실행할 태스크 목록을 유지 관리합니다.

crontab 명령

```
]$ crontab -a fileName
```

cron 대몬(daemon)이 실행할 명령과 단계를 포함하는 crontab 파일을 만듭니다.

crontab 파일을 나열, 수정, 삭제하는 데에도 사용할 수 있습니다.

- crontab 파일 형식에는 다음을 나타내는 필드 6개가 있습니다.
 - MIN: 분 – 0에서 59 사이의 값
 - HOUR: 시간 – 0에서 23 사이의 값
 - DOM: 일 – 1에서 31 사이의 값
 - MON: 월 – 1에서 12 사이의 값
 - DOW: 요일 – 0에서 6 사이의 값
 - CMD: 명령 – 모든 명령 또는 경로

crontab 파일 편집 및 리스팅

```
]$ crontab -e
```

이 명령은 crontab 파일을 루트 사용자 자격으로 편집합니다.

- 필요할 때 되돌릴 수 있도록 `-e`를 입력하기 전에 원본 crontab 텍스트를 새 파일에 복사합니다.

```
]$ crontab -l
```

목록을 보려면 `crontab -l`을 사용합니다.

확인 질문

1. 프로세스를 중지해야 하는 이유는 무엇입니까?
2. `ps -ef | grep [process name]`이 유용한 이유는 무엇입니까?
3. `at` 명령과 `cron` 명령의 차이점은 무엇입니까?

요점



- Linux 시스템은 대몬(daemon), 서비스, 프로그램을 실행합니다.
- `ps` 명령과 `pstree` 명령을 사용하여 실행 중인 프로세스를 표시할 수 있습니다.
- `top` 명령을 사용하여 서버에서 실행되는 프로세스를 검사하고, 리소스 사용률을 검사할 수 있습니다.
- `at` 명령과 `cron` 명령을 사용하여 명령이 실행되는 시간을 예약할 수 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.

