Process utility

**1. 프로세스의 관련 명령어**

* 프로세스 관련 명령어
* (1) ps(process status)
  + 현재 실행중인 프로세스의 상태를 보여주는 명령어이다.
  + **CPU 사용도가 낮은 순서**로 출력한다.
  + 형식: ps [옵션]
  + 옵션 없이 명령어 ‘ps’를 실행하면 자신의 터미널에서 실행되고 있는 프로세스들의 관련 정보만 나타낸다.

|  |  |
| --- | --- |
| **상태** | **설명** |
| **PID** | 프로세스 식별번호 |
| **TTY** | 프로세스와 연결된 터미널 번호 |
| **TIME** | 총 CPU 사용 시간 |
| **CMD** | 실행 명령 |

* + 명령어 ps에서 많이 사용하는 옵션들은 아래와 같다.

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **a** | 현재 실행 중인 모든 프로세스 출력 |
| **e** | 모든 프로세서 정보 |
| **u** | 사용자 이름과 프로세스 시작 시간 출력 |
| **x** | 접속된 터미널뿐만 아니라 사용되고 있는 모든 프로세서들을 출력 |
| **-l** | 자세한 정보 출력 |

* + 명령어 ‘ps’와 ‘grep’을 이용하여 특정 프로세스의 상태 정보를 확인할 수 있다. 명령어 ‘ps ax | grep httpd’는 httpd 프로세스들의 상태를 나타낸다.
  + 상태 설명

|  |  |
| --- | --- |
| **상태** | **설명** |
| **USER** | 프로세스 소유자명(계정명) |
| **PID** | 프로세스 식별번호 |
| **%CPU** | CPU 사용 비율의 추정치 |
| **%MEM** | 메모리 사용 비율의 추정치 |
| **VSZ** | 페이지 단위의 가상메모리 사용량 |
| **RSS** | 실제 사용된 메모리량(Resident Set Size) |
| **TTY** | 프로세스와 연결된 터미널 번호 |
| **STAT** | 현재 프로세스 상태   |  |  | | --- | --- | | **P** | 수행 가능/수행 중 | | **R** | 현재 실행 상태 | | **S** | 잠든 상태, 20초 이상 된 상태 | | **D** | I/O 작업이 끝나기를 기다리면서 정지된 상태(입출력 작업 대기 상태) | | **T** | 정지된 상태(suspend), 일시 정지 상태 | | **Z** | 좀비(Zombie) 프로세스, **작업종료 후 부모 프로세스로부터 회수되지 않아 메모리에 적재되어 있는 상태** | |
| **START** | 프로세스 시작 시간 |
| **TIME** | 총 CPU사용 시간 |
| **COMMAND** | 실행 명령 |

* + 좀비 프로세스는 kill 시그널을 받아도 종료되지 않는다. 이 프로세스는 이미 현재 프로세스에 대한 모든 정보는 메모리에서 사라졌지만 부모 프로세스가 정상적인 종료 처리를 하지 못해 발생
  + 명령어 ‘ps -l’을 수행한 결과 상태 설명

|  |  |
| --- | --- |
| **상태** | **설명** |
| **F** | 프로세스 플래그로 8진수 값의 합을 나타냄  -01: 주기억장치 내에 있음  -02: 시스템 프로세스  -04: 주기억장치 내에서 잠겼음(Locked)  -10: 스왑(SWAP)되고 있음  -20: 다른 프로세스에 의해 추적되고 있음 |
| **S** | 프로세스 상태를 표시(state)  -R: Running -T: Terminated  -S: Sleep -Z: Stopped  -W: Waiting -I: Intermediated |
| **UID** | 프로세스 소유자의 Username |
| **PPID** | 부모의 프로세스의 PID |
| **C** | 짧은 시간 동안의 CPU 사용률 |
| **PRI** | 실행 우선순위 값, 값이 **작을수록** 우선순위가 높음(100~139) |
| **NI** | NICE 우선순위를 계산할 때 사용되는 nice값(-20~19) |
| **ADDR** | 프로세스의 주기억장치 내의 주소 |
| **SZ** | 프로세스 크기로 단위는 블록 |
| **WCHAN** | 프로세스가 대기나 수면 상태에서 기다리는 사건  기다리는 사건이 실행될 주기억장치 내의 주소를 가리킴  실행중인 프로세스는 공백 |

* + 일반적인 프로세스: PR값은 PR = NI + 20 (priority 값은 100~139이다.)
  + 리얼타임 프로세스: PR값은 PR = -1 – rt\_priority(rt\_priority범위는 1~99까지)(priority값은 1~100)
  + NICE값이 ‘프로세스의 실행 우선순위가 높다’라는 의미는 실행 우선순위가 낮은 프로세스보다 더 **많은 시스템 자원을 할당하게 되므로 실행속도가 빨라지게 된다는 것**을 뜻한다. 즉, **NICE값으로 프로세스의 실행 우선순위를 설정**한다.
* (2) pstree
  + 명령어 ‘pstree‘응 실행 중인 프로세스들을 트리구조로 나타낸다.
  + 형식: pstree [옵션]

|  |  |
| --- | --- |
| **상태** | **설명** |
| **-p** | 프로세스 ID표시 |
| **-n** | 프로세스 ID를 **정렬**해서 표시 |
| **-a** | **명령행 인자를 포함**한 정보를 트리구조에 표시 |
| **-h** | 현재 프로세스와 해당 프로세스의 부모 프로세스를 음영으로 강조해서 표시 |

* (3) jobs
  + 작업이 중지된 상태(suspend), 백그라운드로 진행 중인 상태(background), 변경되었지만 보고되지 않은 상태 등을 표시한다.
  + 형식: jobs [옵션] [작업번호]

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-l** | 프로세스 그룹 ID를 state 필드 앞에 표시 |
| **-n** | 프로세스 그룹 중 대표 프로세스 ID 표시 |
| **-p** | 프로세스 ID를 한 행 씩 표시 |

* + 백그라운드로 실행 중인 프로세스를 확인한다. [숫자]는 ‘작업번호’이다.
  + 출력된 목록에서 +는 현재 작업 실행, -는 앞으로 실행될 작업(대기 상태 작업)을 나타낸다.
  + 작업의 상태 값은 다음과 같다.

|  |  |
| --- | --- |
| **상태** | **설명** |
| **Running** | 작업이 일시 중단되지 않았고 종료하지 않고 계속 진행 중 |
| **Stopped** | 작업이 일시 중단 |
| **Done** | 작업이 완료되어 0을 반환하고 종료 |
| **Done(code)** | 작업이 정상적으로 완료되었으며, 0이 아닌 코드를 반환 |

* (4) bg와 fg
  + 포그라운드에서 백그라운드로의 전환
    - 명령어 bg는 현재 실행 중인 프로세스를 백그라운드 작업으로 전환한다.
    - 형식: bg %작업번호 또는 bg 작업번호
    - 작업이 빨리 끝날 것이라고 예상하고 실행했지만 시간이 지나도 끝나지 않을 때 또는 현재 작업은 놔둔 상태로 다른 작업을 실행하고 싶을 때 백그라운드로 전향해서 작업을 진행하는 경우가 있다.
    - 포그라운드에서 백그라운드로 전환 시 **먼저 실행중인 작업을 일시 정지시킨 후**(Ctrl + Z), bg 명령을 수행한다.
    - **명령어 실행을 백그라운드 작업으로 실행**하고자 할 때는 **명령어 뒤에 ‘&’**를 붙여주면 된다.
  + 백그라운드에서 포그라운드로의 전환
    - 명령어 fg는 현재 백그라운드로 실행 중인 명령어를 포그라운드로 전환한다.
    - 형식: fg %작업번호 또는 fg 작업번호
    - 작업번호를 부여하지 않으면 현재 수행 중인 작업을 포그라운드로 전환한다.
    - 포그라운드 작업을 종료하려면 Ctrl + C를 입력한다.
* (5) kill
  + 프로세스를 종료시킨다.
  + 형식: kill [옵션] [시그널번호 또는 시그널이름] [PID 또는 %작업번호]

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-l** | 시그널 종류 나열 |
| **-s 시그널번호**  **-s 시그널명** | 전달한 시그널의 종류를 지정 |
| **-1** | -HUP, 프로세스를 재시작 |
| **-9** | 프로세스를 강제로 종료 |

* (6) killall
  + 같은 데몬의 여러 프로세스를 한 번에 종료시킬 때 사용된다.
  + 프로세스명으로 연관된 프로세스들을 종료시킨다.
  + 형식: killall [옵션] 프로세스명

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-g** | **그룹을 지정하여 프로세스를 종료**  같은 프로세스 그룹에 속한 여러 프로세스가 발견되더라도 시그널은 그룹별로 한 번만 보내짐 |
| **-i** | 프로세스 종료 전 확인 메시지 표시 |
| **-l** | 시그널 목록 표시 |
| **-v** | 상세 정보 표시 |

* + 명령어 ‘killall -l’은 시그널 옵션들을 확인할 수 있다.
* (7) nice
  + 프로세스 사이의 우선순위를 확인하고 **우선순위를 변경할 수 있는 명령어**이다.
  + NI값을 조정하여 **프로세스의 우선순위를 변경**한다.
  + 형식: nice [옵션] 프로세스명

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-n 조정수치**  **-조정수치**  **--adjustment 조정수치** | 명령의 우선권에 조정수치를 더함(양수값: -, 음수 값: --) |

* + 조정할 수 있는 NI값의 범위는 -20(가장 높은 우선권) ~ 19(가장 낮은 우선권)이다
  + 우선순위 0의 값을 가지며 값이 작을수록 우선순위가 높다.
  + 옵션 -n을 사용하지 않으면 디폴트는 10을 사용한다.
  + 조정수치가 생략되면 명령의 우선권은 10만큼 증가한다.
  + 일반 사용자는 nice값을 증가시킬 수밖에 없으며, root는 nice값을 감소시켜 우선순위를 높일 수 있다.
  + 명령어 ‘nice -10 bash’는 bash 프로세스 NI값을 10만큼 증가시키는 것이다. 값을 증가시킨다는 것은 우선순위를 낮추는 것이다. 우선 순위를 높이는 명령어는 ‘nice --10 bash’이다. 이것은 NI값을 -10만큼 감소시켜 우선순위를 높인다.
* (8) renice
  + **이미 실행중인 프로세스의 우선순위를 변경**한다.
  + 프로세스 소유자와 루트 권한자만이 명령을 내릴 수 있다.
  + 형식: renice [옵션] NI값 PID

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-p** | 프로세스 ID 지정 |
| **-u** | 사용자명 지정 |
| **-g** | 프로세스의 GID 지정 |

* + 우선순위 변경 시 프로세스 ID, 사용자 이름, 프로세스 그룹 ID를 사용한다.
  + 명령어 nice는 기존 NI값을 증감하지만 명령어 renice는 지정한 NI값을 설정한다.
  + nice는 양수 값은 -를, 음수 값은 --를 사용하지만, renice는 양수 값에 -를 사용하지 않는다.
  + 명령어 ‘renice -20 3241’는 프로세스 3241의 NI값을 -20으로 변경한다.
* (9) top
  + 리눅스 시스템의 운영 상태를 실시간으로 모니터링하거나 프로세스 상태를 확인할 수 있다.
  + 형식: top [옵션] [PID]

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-d 시간** | 실시간 화면 출력 시간 지정(초 단위) |
| **-p PID** | 모니터할 프로세스 ID(PID) 지정 |

* + 필드 설명

|  |  |
| --- | --- |
| **상태** | **설명** |
| **PID** | 프로세스 식별자 번호 |
| **USER** | 프로세스 소유자 |
| **PR** | 프로세스 우선순위 |
| **NI** | 프로세스의 NICE값 |
| **VIRT** | 해당 프로세스가 사용하는 가상 메모리의 총량(Virtual Image(kb)) |
| **RES** | 해당 프로세스의 물리적 메모리 사용량(Resident Size(kb)) |
| **SHR** | 해당 프로세스에 의해 사용된 공유 메모리 총량(Shared Mem Size(kb)) |
| **S** | 해당 프로세스의 상태(state)  - D: 중단될 수 없는 Sleep 상태(uninterrupted sleep)  - R: 실행 중인 상태(Running)  - S: 휴먼상태(Sleep)  - T: Trace 되거나 정지된 프로세스  - Z: 좀비 프로세스 |
| **%CPU** | 해당 프로세스의 CPU 사용률 |
| **%MEM** | 해당 프로세스의 메모리 사용률 |
| **TIME+** | 100분의 1초 단위로 나타내는 CPU 사용 시간 |

* + 기본으로 5초에 한 번씩 실시간으로 시스템 상태 정보가 갱신된다.
  + top이 실행된 상태에서 다양한 명령을 입력하여 프로세스의 상태를 출력하거나 제어할 수 있다.

|  |  |
| --- | --- |
| **h 또는 ?** | 도움말 확인(help) |
| **t** | 두 번째와 세 번째 행 정보인 CPU와 프로세스 정보를 보여주거나 감춤(토글) |
| **i** | 좀비 프로세스 정보를 보여주거나 감춤(토글) |
| **n 수치** | 지정된 개수만큼의 프로세스만을 화면에 표시 |
| **r 조정치** | renice값을 변경하여 실행 우선순위 조정 |
| **k PID** | 지정된 PID를 종료(kill -9 PID와 동일) |
| **q 또는 Ctrl+C** | top실행 종료 |

* (10) nohup(no hang up)
  + **프로세스가 중단되지 않고 백그라운드로 작업을 수행할 수 있게 한다.**
  + 형식: nohup 명령어(문) &
  + 사용자가 로그아웃 하거나 작업 중인 터미널 창이 닫혀도 실행 중인 프로세스를 백그라운드 프로세스로 계속 작업할 수 있도록 한다.
  + 용량이 큰 데이터 압축 해제와 같은 실행 시간이 오래 걸리는 프로세스들에 대해 nohup으로 처리하여 작업하면 작업 중단 없이 해당 업무를 완료할 수 있다.
  + 백그라운드로 실행될 수 있도록 명령행 뒤에 ‘&’를 명시한다.
* (11) tail
  + 파일의 마지막 행을 기준으로 지정한 행까지 파일 내용의 일부를 출력한다.
  + 기본값으로 마지막 10줄을 출력한다.
  + 형식: tail [옵션] [파일명]

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-f** | 파일의 10줄을 출력하고 파일의 내용을 실시간으로 출력 |
| **-n N** | N 개수만큼의 라인을 출력 |

**2. 스케줄링과 cron**

* **주기적으로 반복되는 일을 자동적으로 실행될 수 있도록 설정한다.**
* **스케줄링 데몬(서비스)은 “crond”**이며, 관련 파일은 “/etc/crontab”이다.
* 파일 /etc/crontab은 7개의 필드로 구성되어 있다.
* 명령어 crontab은 사용자가 주기적인 작업을 등록할 수 있게 한다.
  + 형식: crontab [옵션] 파일명

|  |  |
| --- | --- |
| **옵션** | **설명** |
| **-l** | crontab에 설정된 내용 출력 |
| **-e** | crontab을 작성하거나 수정(edit) |
| **-r** | crontab 내용 삭제(remove) |
| **-u** | 특정 사용자의 일정 수정 |

**1. minute:** 0-59

**2. hour:** 0-23

**3. day of month:** 1-31

**4. month:** 1-12 OR jan, feb, mar, apr …

**5. day of week:** 0-6(Sunday = 0 or 7) OR sun, mon, the, wed, thu, fri, sat

**6. 시간 설정:** 백업 스크립트 파일 /etc/backup.sh를 매주 화요일과 목요일 오전 4시에 실행하려면

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **분(min)** | **시(hour)** | **날(day)** | **달(month)** | **요일** | **명령어** |
| \* | 4 | \* | \* | 2,4 | /etc/backup.sh |

**위를 참고하여 /etc/crontab에 \* 4 \* \* 2,4 /etc/backup.sh**를 추가하면 된다.