

## 6장

### 작업 실행 시 작업 실행 상황 확인

`--dryrun` : 어떤 command들이 실행되고 있는지 확인

`--verbose` : command와 출력 결과를 같이 확인

### 모든 작업 완료 전에 출력

`--ungroup` / `-u` : 즉시 결과 출력

ungroup은 빠르지만 실행 결과가 섞이는 경우가 있다.

-> `--linebuffer` 로 해결 가능하다.

`--keep-order` `--line-buffer` 같이 사용시  
하나의 작업이 완료된 후에 다음 작업을 출력한다.

### 출력을 file로 저장

`--files` : 기본적으로 /tmp 에 출력 결과를 저장. 저장 위치와 파일 이름을 출력

`--tmpdir` 위치 / `RMPDIR=위치` 로 저장 위치를 설정 가능하다.

### 출력 결과를 출력과 파일 두 방식으로 모두

`--results`

---

## 7장

### Number of simultaneous jobs

`-j` / `--jobs` : 최대 동시 실행 작업 수

### Shuffle job order

`--shuf` : 작업의 순서를 변경

### Interactivity

`--interactive`: 작업마다 실행 여부를 입력

???

-tty : 대화형 명령 실행 가능

## terminal for every job

--tmux

각 작업을 별도의 창으로 실행할 수 있다.

--tmuxpane 화면을 분할해서 작업 실행

--fg 작업을 포그라운드에서 실행, 각 분할 화면을 즉시 확인 가능

## Timing

--delay 초: 작업 시작 사이에 지연 시간을 두고 실행하는 기능

--timeout: 작업이 지정된 시간 내에 완료되지 않으면 종료

초, 분, 시, 일 단위로 표현 가능: 100000 == 1d3.5h16.6m4s

## Progress information

--progress:작업의 진행 상황 표시

-> 현재 실행 중인 작업의 수, 완료된 작업의 수, 시작된 작업의 비율, 각 작업의 평균 실행 시간.

--eta: 완료된 작업을 기준으로 모든 작업이 완료되는 시간 예측

--bar : 작업 진행 상황을 진행 바로 표시

## Logfile

--joblog 위치 : 실행한 작업들의 로그 파일을 생성

## Resume jobs

joblog 로 생성된 로그파일로 작업 재시작이 가능하다.

--resume: 이미 실행된 작업들은 무시되고, 아직 실행되지 않은 작업들만 실행

--resume-failed: 이전에 실패한 작업만 재실행

--retry-failed: 실패한 작업을 자동으로 재시도

## Termination

--halt (soon/now),(fail/done)=\_

now: 조건 충족 시 바로 작업을 종료, 추가 작업 시작하지 않음

soon: 현재 실행 중인 작업들이 끝날 때 까지 대기, 새로운 작업 시작하지 않음.

fail은 실패

done은 성공(완료)

개수 or 비율

`--halt soon,fail=1:`

- 하나의 작업이 실패하면 나머지 작업을 중지하고 더 이상 작업을 시작하지 않습니다.

`--halt soon,fail=20%:`

- 지정된 비율(20%)만큼 작업이 실패하면 더 이상 작업을 시작하지 않습니다.

## Retry Failing Commands

`--retries`: 명령어가 실패했을 때 재시도를 지정하는 데 사용. 특정 작업이 실패했을 때 최대 지정된 횟수만큼 재시도

`--termseq`: 작업을 종료할 때 보낼 신호의 순서를 제어

`--termseq` 신호,대기시간,신호,대기시간....

## Limit the Resources

`--nice`: 작업의 우선순위설정

`--load`: 시스템의 부하를 기준으로 새 작업을 실행할지 결정

`--load 100%`는 시스템 부하가 CPU 코어 수의 100% 이하일 때만 작업을 시작

`--noswap`: 시스템이 스왑(Swap)을 사용하는지 확인하여, 스왑을 사용 중일 때는 작업을 실행하지 않음.

`--memfree`: 사용 가능한 메모리를 기준으로 작업을 실행,메모리가 부족해지면 실행 중인 작업을 종료

`--memfree 1G`는 시스템에 1GB 이상의 여유 메모리가 있을 때만 작업을 시작

---

# 8장

ssh 세팅이 되어 있어야 한다.

## sshlogin

`--sshlogin / -S` : 실행할 서버를 지정하여 그 서버에서 명령을 실행

## Multiple servers

-s 옵션을 여러 번 사용

```
parallel -S $SERVER1 -S $SERVER2
```

심표로 서버 구분

```
parallel -S $SERVER1,$SERVER2
```

--sshloginfile : 파일로 저장된 서버 목록으로 ssh 연결

```
parallel --sshloginfile severfile
```

number/서버 로 해당 서버에서 cpu 코어 수를 number로 제한할 수 있다.

## Transfer files

--transferfile : 원격 서버에서 처리할 파일을 전송

--return {}: 결과 파일을 로컬로 전송

--cleanup: 파일을 전송한 후, 원격 서버에서 해당 파일을 자동으로 삭제

## Working dir

--workdir: 명령어를 원격 서버에서 실행할 작업 디렉토리를 지정

~가 기본 작업 디렉토리

## Run the same commands on all hosts

--onall: 모든 호스트에서 동일한 명령을 실행

---

# 9장

입력 파일을 여러 블록으로 나누어 각 작업에 전달

--pipe : 입력 파일을 블록 단위로 나누어 병렬로 처리할 수 있습니다.

--block 크기: 블록의 크기를 변경

--round-robin: 작업을 균등하게 분배할 때 사용