MCNL 여름 방학 스터디 6주차

MCNL

Yunmin Go

School of CSEE

HANDONG GLOBAL UNIVERSITY

Schedule

- 1주차: Segment tree, Union-Find, Red-black tree, Trie
- 2~3주차: Trie 기반 자동 완성, Red-Black Tree 기반 Map 구현
- 4주차: Malloc 구현
- 5주차: Network Programming study, Simple Search Engine 구현
- 6주차: Network Programming study, Simple Git 구현

- 코딩 테스트 1주일에 3개씩

Simple Git 구현

- 아래의 기능이 가능한 Git 서버와 클라이언트를 구현하시오.
  - 서버 실행 시 등록해 놓은 디렉토리(Repository)에 대해서만 작업 수행
    - Branch 관련 기능은 제공하지 않아도 됨

```
$ ./simplegit_server 8080 /home/yunmin/repository
```

  - 서버의 IP와 Port번호는 `simplegit_addr.txt` 파일에 저장되어 있으며, 클라이언트는 이를 참조하여 서버에 접속함

```
$ cat simplegit_addr.txt
192.168.10.10  # IP
8080           # Port
```

Simple Git 구현

- 아래의 기능이 가능한 Git 서버와 클라이언트를 구현하시오.
  - 클라이언트는 init을 이용하여 현재 디렉토리에 대해 새로운 Repository를 생성할 수 있음

```
$ ./simplegit init
```

  - 클라이언트의 로컬 Repository에서 파일을 추가하거나 수정한 이후에는 commit을 수행할 수 있음
    - 클라이언트의 로컬 Repository에서 추가하거나 수정한 모든 파일에 대해서 자동으로 Tracking이 되어야 함 (index에 자동 추가됨)

```
$ ./simplegit commit –m "first commit"
```

    - Commit 시 수정된 파일의 추가 및 삭제된 부분을 요약해서 표시함 (git 참고)

Simple Git 구현

- 아래의 기능이 가능한 Git 서버와 클라이언트를 구현하시오.
  - 클라이언트는 Commit 이후 서버의 Repository에 Push할 수 있음

    ```
    $ ./simplegit push
    ```

    - Push 하는 파일이 서버의 Repository에 있는 파일과 Conflict이 발생할 경우 서버는 Push를 Reject할 수 있으며 Conflict된 내용을 파일에 기록하여 알려줌

  - 클라이언트는 Pull을 이용하여 자신의 로컬 Repository를 서버의 최신 Repository로 업데이트 할 수 있음

    ```
    $ ./simplegit pull
    ```

    - 이 숙제에서는 pull 기능이 git의 clone 기능도 포함함

  - 위에서 언급한 것 이외의 기능은 자유롭게 추가해도 됨

Simple Git 구현

- 아래의 시나리오가 가능해야 함

순서	Client #1	Client #2
1	`init`	
2	sample.txt 생성 (임의의 내용 추가)	
3	commit	
4	push → 성공	
5		pull (=clone)
6		sample.txt에 한 줄 추가
7		commit
8		push → 성공
9	sample.txt에 한 줄 추가	
10	Commit	
11	Push → 실패	