

## 목차

1. [과제 개요](#)
2. [기능](#)
3. [상세 설계](#)
4. [실행 결과](#)

### 1. 과제 개요

리눅스 시스템 상에서 사용자가 원하는 파일 또는 디렉토리를 옵션에 따라 백업하거나 백업한 파일을 다시 복원 및 삭제하고, 백업 상태를 관리하는 ssu\_backup 프로그램을 작성하였다. 이를 위해 리눅스 시스템에서 제공하는 여러 시스템 자료구조와 시스템 및 라이브러리 콜을 이용하였으며, 백업 파일 관리를 위한 연결 리스트를 구현하여 백업된 파일들을 관리하였다. 또한 ssu\_backup 프로그램에서 사용할 수 있는 내장명령어는 backup, remove, recover, list, help가 있으며, 각각의 내장 명령어는 fork()와 exec() 시스템 콜을 호출함으로써 새로운 프로세스에서 실행된다.

### 2. 기능

ssu\_backup 프로그램은 입력 받은 명령행 인자에 따라 다른 프로세스를 실행시킨다.

```
// main.c
// 내장 명령어에 따라 작업 수행하고, 만약 잘못된 명령어가 입력된 경우에는 에러 처리 후 프로그램 비정상 종료
if(!strcmp(argv[1], command[0]))
    runCmd(command[0], arglist);
else if(!strcmp(argv[1], command[1]))
    runCmd(command[1], arglist);
else if(!strcmp(argv[1], command[2]))
    runCmd(command[2], arglist);
else if(!strcmp(argv[1], command[3]))
    runCmd(command[3], arglist);
else if(!strcmp(argv[1], command[4]))
    help(arglist[2]);
else{
    fprintf(stderr, "Usage: %s <backup|remove|recover|list|help> [option] ...\n", argv[0]);
    exit(1);
}

// command.c
char *command[NUM_CMD] = {
```

```

    "backup",
    "remove",
    "recover",
    "list",
    "help"
};

void runCmd(char *cmd, char **arglist)
{
    pid_t pid;
    int status;

    // 프로세스 생성 실패
    if((pid = fork()) < 0){
        fprintf(stderr, "fork error for %s command\n", cmd);
        exit(1);
    }

    // 자식 프로세스: cmd 에 해당하는 프로그램 실행
    // /home/backup 디렉토리에 접근해야 하므로 root 권한으로 실행
    // sudo ./command [path] [option] ...
    else if(pid == 0){
        char *temp = (char *)malloc(sizeof(char *) * (strlen(arglist[1]) + 3));

        // arglist[0]을 ./ssu_backup 에서 sudo 로 변경해줌
        strcpy(arglist[0], "sudo");

        // 내장 명령어에 해당하는 파일의 경로를 저장해줌
        strcpy(temp, ".");
        strcat(temp, arglist[1]);
        strcpy(arglist[1], temp);

        execvp("sudo", arglist);
        exit(0);
    }

    // 부모 프로세스
    else{
        wait(&status);

        // 자식 프로세스가 정상적으로 종료되었으면 true 리턴
        // 만약 자식 프로세스가 비정상적으로 종료되었다면 부모 프로세스도 비정상 종료 처리
    }
}

```

```

        if(!WIFEXITED(status))
            exit(1);
    }
}

```

경로를 필요로 하는 내장 명령어들은 입력 받은 경로를 절대 경로로 변환한 후, 해당 경로가 올바른지 검사한다. 또한 옵션을 필요로 하는 내장 명령어들은 옵션을 추출하고, 입력 받은 경로와 옵션이 서로 올바른지 검사를 한다. 이와 같은 함수는 다음과 같다.

```

int processPath(char *path, char *fullPath, char *name); // 입력 받은 경로를 통해 절대 경로와 파일 및
디렉토리의 이름을 추출
int checkPath(char *path, char *name); // 해당 경로가 올바른 경로인지 검사
int getOpt(char *opt, int argc, char *argv[]); // 옵션 추출하여 opt 에 저장
int checkOpt(mode_t mode, char opt); // 압력받은 경로에 맞는 올바른 옵션이 입력되었는지 확인

```

각 명령어들마다 올바른 경로 및 옵션이 입력되었다면, 각 명령어에 해당하는 작업을 수행한다.

#### 1) backup <PATH> [OPTION] ...

백업할 경로 PATH를 입력 받아 해당 파일 또는 디렉토리를 백업 디렉토리 내에 백업하는 명령어이다. 옵션에 따라 파일 또는 디렉토리를 백업할 수도 있으며, 이미 백업 본이 존재하는 파일을 백업할 수도 있다. 백업된 파일 및 디렉토리는 백업 디렉토리(/HOME/backup) 내에 생성된 백업 시간을 버전으로 하는 디렉토리에 저장된다.

```

// backup.c (backup 프로그램 실행)
int isBackuped(BackupDir *backupDirNode, char *originPath, char *name); // 백업하려는 파일이 백업
디렉토리 내에 백업되어 있는 상태인지 검사
void backupFile(char *parentPath, char *name, char *subPath, char *date); // 파일 백업
void backupDir(char *originPath, char *subPath, char *date, int (*filter)(const struct dirent *), int allowDup); //
디렉토리 백업

```

#### 2) remove <PATH> [OPTION] ...

입력 받은 PATH에 대해 백업 디렉토리에 백업되어 있는 백업 파일 또는 디렉토리를 삭제하는 명령어이다. 옵션에 따라 파일 또는 디렉토리를 여러 개 삭제할 수도 있으며, 원하는 파일을 골라서 삭제할 수도 있다. 또한 명령어 실행 후 해당 버전 디렉토리 안에 아무 파일이 없다면 해당 버전의 디렉토리를 삭제한다.

```

Dir *canRemove(char *originPath, char *name, int *isReg); // PATH 에 해당하는 백업 파일 또는
디렉토리가 있는지 확인
void removeFile(Dir *removeDirList, char *parentPath, char *name, int removeAll, char *date); //
백업디렉토리 내 파일 삭제
void removeDir(Dir *removeDirList, char *originPath, int opt, char *date); // 백업 디렉토리 내 디렉토리의
하위 폴더 또는 디렉토리 삭제

```

#### 3) recover <PATH> [OPTION] ...

입력 받은 PATH에 대해 백업 디렉토리에 백업되어 있는 백업 파일 또는 디렉토리를 복구하는 명령어이다. 옵션에 따라 파일 또는 디렉토리를 여러 개 복구할 수도 있으며, 최근에 복구된 백업 본을 복구할 수도 있고, 새로운 경로로 백업 본을 복구할 수도 있다. 또한 복구한 백

업 본은 백업 디렉토리 내에서 삭제되며, 해당 버전 디렉토리 안에 아무 파일이 없다면 해당 버전의 디렉토리는 삭제된다.

```
Dir *canRecover(char *originPath, char *name, int *isReg); // PATH 에 해당하는 복구할 수 있는 파일 또는
디렉토리가 있는지 확인
void makeDir(char *newPath); // 복구한 파일은 저장할 경로가 존재하지 않는다면, 해당 경로 생성
RecoverFileList *searchRecoverable(Dir *recoverDirList, char *name); // 복구 가능한 파일 리스트들을
추출
int selectRecoverFile(RecoverFileList *recoverFileList); // 복구 가능한 파일들 중 복구할 파일 선택

void recoverFile(RecoverNode *file, char *newPath, char opt, char *date); // 파일 복구
void recoverDir(Dir *recoverDirList, char *newPath, int opt, char *date); // 디렉토리 내부의 파일 또는
디렉토리 복구
```

#### 4) list [PATH]

백업 디렉토리 내의 파일 및 디렉토리 구조를 트리 형태로 출력하고 사용자 입력에 따라 명령을 수행하는 명령어이다. 사용자 입력으로는 특정 경로에 대해 remove 명령어와 같이 처리하는 rm, recover 명령어와 같이 처리하는 rc, 원본 파일을 수정할 수 있는 vi(m), 프로그램을 종료하는 exit가 있다.

#### 5) help [COMMAND]

ssu\_backup 프로그램의 내장 명령어에 대한 설명을 출력하는 명령어이다. COMMAND 입력 시에는 COMMAND에 해당하는 명령어의 설명을 출력하고, help만 입력되었을 때에는 모든 내장 명령어에 대한 설명을 출력한다. 이를 위해 usage라는 디렉토리에 각각의 내장 명령어의 설명을 저장해둔 후, 필요할 때마다 open()을 통해 해당 설명 파일을 열고, 파일의 내용을 프롬프트에 출력하였다.

usage
≡ backup.txt
≡ help.txt
≡ list.txt
≡ recover.txt
≡ remove.txt

```
// 각 명령어의 사용법에 해당하는 파일명(fname) 저장
if(!strcmp(argv[1], command[0]))
    sprintf(fname, "./usage/%s.txt", command[0]);
else if(!strcmp(argv[1], command[1]))
    sprintf(fname, "./usage/%s.txt", command[1]);
else if(!strcmp(argv[1], command[2]))
    sprintf(fname, "./usage/%s.txt", command[2]);
else if(!strcmp(argv[1], command[3]))
    sprintf(fname, "./usage/%s.txt", command[3]);
else if(!strcmp(argv[1], command[4]))
    sprintf(fname, "./usage/%s.txt", command[4]);
else{
    fprintf(stderr, "Usage: help [backup|remove|recover|list|help]\n");
    exit(1);
}
```

```

if((fd = open(fname, O_RDONLY)) < 0){
    fprintf(stderr, "open error for %s\n", fname);
    exit(1);
}

```

### 3. 상세 설계

내장 명령어	backup	remove	recover	list	help
명령어 구현	○	○	△	×	○

main.c에서 호출한 runCmd() 또는 help()를 통해 내장 명령어는 새로운 프로세스를 생성 및 실행함으로써 각각의 명령어들이 수행된다. 모든 내장 명령어들은 순서는 조금씩 다르지만 전반적인 흐름은 다음과 같다.

- ① /home 디렉토리 내에 /backup 디렉토리가 없다면 생성한다.
- ② 백업 디렉토리(/home/backup) 내에 로그 파일(ssuback.log)이 없다면 생성한다.
- ③ 현재 프로세스가 실행되고 있는 경로(exePATH), 홈 디렉토리 경로(homePATH), 백업 디렉토리 경로(backupPATH), 로그 파일의 경로(logPATH)를 저장한다.
- ④ 입력 받은 명령행 인자를 통해 내장 명령어 프로세스를 실행시킨다.
- ⑤ 올바른 경로가 입력되었는지 확인한다.
- ⑥ 올바른 옵션이 입력되고, 경로와 잘 짝 지었는지 확인한다.
  - i. 옵션 중 -d, -r을 사용했다면, 입력 받은 경로는 무조건 디렉토리의 경로
  - ii. 옵션 중 -d, -r을 사용하지 않았다면, 입력 받은 경로는 무조건 파일의 경로
- ⑦ 백업 디렉토리(/home/backup)에 백업된 파일들의 정보를 연결리스트에 저장한다.
- ⑧ 백업 본의 정보를 담은 연결리스트를 바탕으로 내장 명령어들을 실행한다.

모든 명령어를 실행하기 전에 백업 디렉토리(/home/backup) 내부의 백업 본들의 버전과 로그 파일(ssubak.log)의 내용을 비교하며, 백업 파일을 관리하는 연결 리스트를 생성한다. 이 때 해당 연결리스트의 노드는 다음과 같다.

```

// /home/backup 내부에 저장된 파일
typedef struct BackupDir{
    char dirName[LEN_DATE + 1]; // 백업 디렉토리명 (백업 시간)
    char *parentPath; // 원본 상위 경로
    char *backupPath; // 백업 디렉토리 내에서 파일이 저장된 상위 경로
    struct BackupDir *next; // 바로 이전에 백업된 디렉토리로, dirName 과 연동된 파일 경로가 동일함
}BackupDir;

// /home/user 내부에 저장된 파일
typedef struct Dir{
    char dirName[MAX_PATH + 1]; // 홈 디렉토리 내에 있는 실제 이름

```

```

struct Dir *subDir; // 하위 디렉토리 중 첫 번째 디렉토리 (오름차순)
struct Dir *siblingDir; // 동일한 위치의 디렉토리 중 다음 디렉토리 (오름차순)
struct BackupDir *head; // originPath 와 연동된 가장 최신의 백업 디렉토리
struct BackupDir *tail; // originPath 와 연동된 가장 과거의 백업 디렉토리
}Dir;

```

Dir \*listHead; // 모든 백업 파일에 대해 로그 파일 내용과 대조하여 파일들에 대한 백업 상태 정보를 저장한 링크드 리스트

구조체 Dir은 원본 경로(/home/user) 내의 디렉토리들의 경로를 연결 리스트로 나타낼 때 사용되는 노드로, 각 노드들은 자신의 하위 디렉토리 혹은 같은 위치에 존재하는 형제 디렉토리를 가리킬 수 있다. 이 때 형제 노드들은 오름차순으로 저장되어 있으며, 각 노드는 백업 본과 연동된 디렉토리를 나타낸다.

▼ assign

- ▼ a
  - ▼ b
    - ≡ c.txt
  - ≡ a.txt
  - ≡ c.txt
- ▼ b
  - ≡ a.txt

▼ backup

- ▼ 240331103713
  - ▼ b
    - ≡ c.txt
    - ≡ a.txt
    - ≡ c.txt
- ▼ 240331103729
  - ≡ a.txt
  - ≡ subak.log

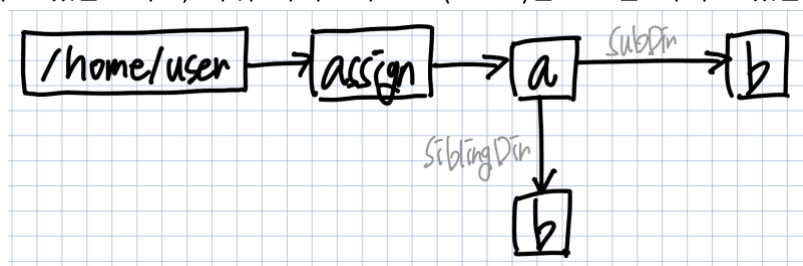
```

240331103713 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331103713/a.txt"
240331103713 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331103713/c.txt"
240331103713 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331103713/b/c.txt"
240331103729 : "/home/ubuntu/assign/b/a.txt" backed up to "/home/backup/240331103729/a.txt"

```

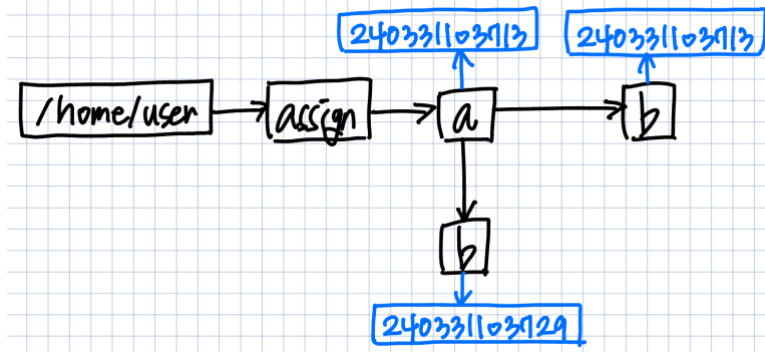
사용자의 홈 디렉토리와 백업 디렉토리의 상태와 로그 파일이 다음과 같을 때, /home/backup/240331103713 내부의 백업 본들은 /home/ubuntu/assign/a와 연동되었음을 알 수 있으며, /home/backup/240331103729 내부의 백업 본들은 /home/ubuntu/assign/b와 연동되었음을 알 수 있다.

따라서 원본 경로(/home/ubuntu/..) 내의 디렉토리의 경로를 구조체 Dir를 통해 연결 리스트로 나타내면 다음과 같다. 다음은 디렉토리의 이름(dirName)과 하위 디렉토리(subDir), 형제 디렉토리(siblingDir)를 나타낸 그림이며, 이 때 /home/ubuntu/assign/a의 형제 노드(siblingDir)는 a.txt를 가지고 있는 b이고, 하위 디렉토리 노드(subDir)는 c.txt를 가지고 있는 b이다.



백업 본들이 저장된 백업 디렉토리(/home/backup/...)와 연동된 모든 원본 경로(/home/user/...)

에 대한 연결 리스트가 생성되었다면, BackupDir 구조체를 이용하여 각 원본 경로들에 연동되어 있는 백업 본들에 대한 정보를 Dir 구조체의 head 및 tail에 연결해준다. 연결 리스트를 생성해 줄 때에는 백업 디렉토리(/home/backup) 내 버전 디렉토리들을 오름차순으로 추출하여, 각 버전에 해당하는 로그 파일(ssubak.log)의 내용을 통해 노드를 생성하고 연결리스트의 가장 끝(tail)에 연결해주기 때문에 자연스럽게 백업 본들은 오름차순으로 저장된다. 즉, 끝으로 갈 수록(tail에 가까워질 수록) 최신 버전이다. 다음은 위에서 생성한 연결 리스트에 연동된 백업 파일들을 연결한 것으로, 구조체의 멤버 변수 중 해당 백업 본 백업 시각(백업 디렉토리 내 백업 버전을 나타내는 디렉토리명)인 dirName을 나타낸 것이다.



이 때 listHead는 생성한 백업 본들의 관리 연결 리스트의 시작 노드인 /home/user를 가리킨다. 이러한 연결 리스트의 생성은 filepath.c에 정의되어 있는 아래의 함수들을 통해 실행된다.

```
// filepath.c
// 모든 백업 파일에 대해 로그 파일 내용과 대조하여 파일들에 대한 백업 상태 정보를 링크드 리스트에 저장
Dir *createBackupList();

// backupDir 과 연동되어 있는 원본 경로 디렉토리 노드가 listHead 연결리스트 내 존재하는지 확인하고
// 존재하지 않는다면 노드를 생성 및 연결하고, 연동되어야 할 원본 경로 디렉토리 리턴
static Dir *insertDirNode(Dir **listHead, char *path);

// 해당 디렉토리(dir)와 연동되어 있는 백업 디렉토리(/home/backup) 내 디렉토리(backupDir)를 dir 의
// 연결리스트에 연결
// 즉, dir 은 해당 디렉토리와 연결되어 있는 backupDir 노드들 간의 연결리스트의 헤더(더미) 노드
static void insertBackupDirNode(Dir **dir, BackupDir *backupDir);
```

다음은 각 내장 명령어들 중 구현한 명령어어들의 흐름도와 각 명령어어에서 사용 가능한 옵션의 구현 여부 등을 기술했다.

⑩ Backup 명령어

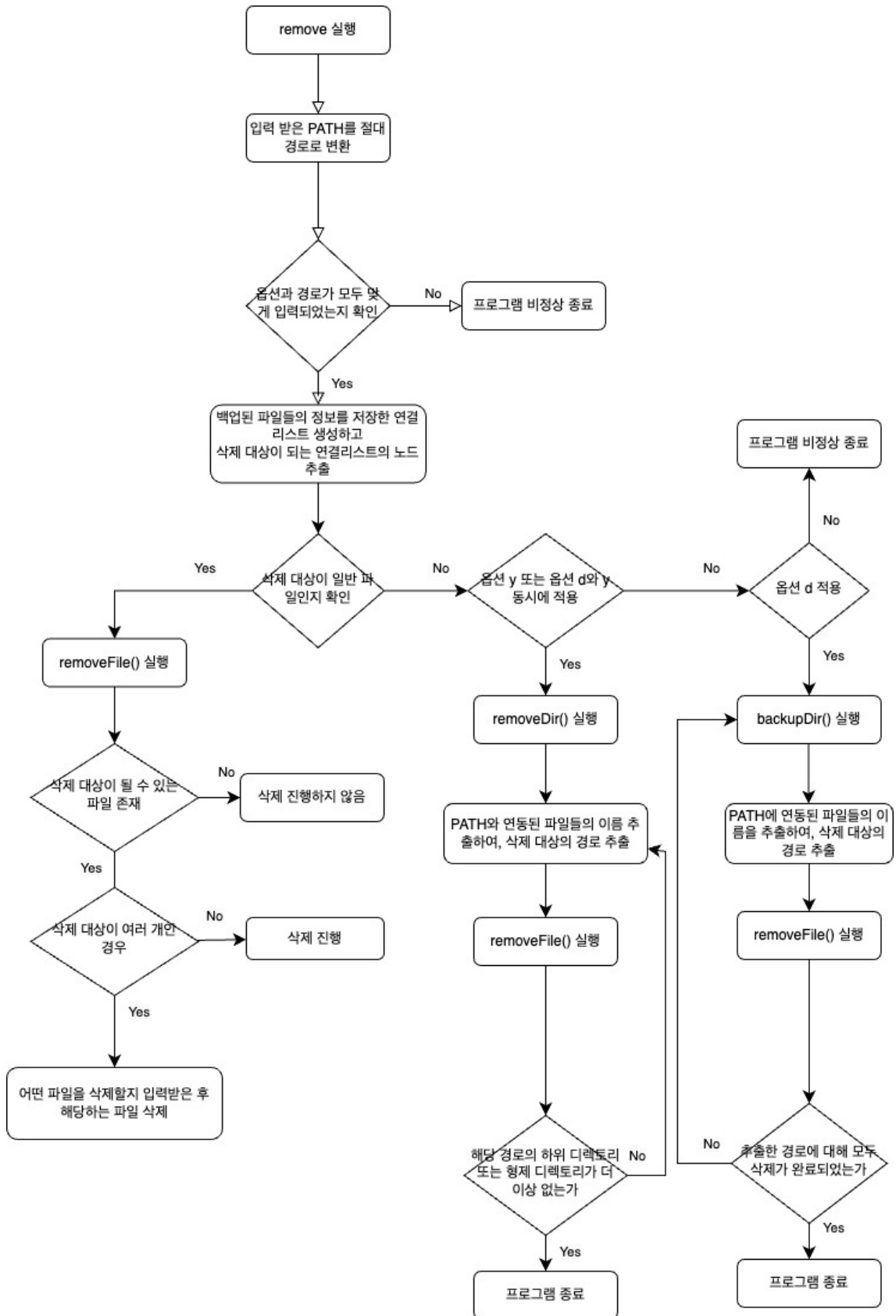
명령어 구현	-d 옵션	-r 옵션	-y 옵션
○	○	○	○





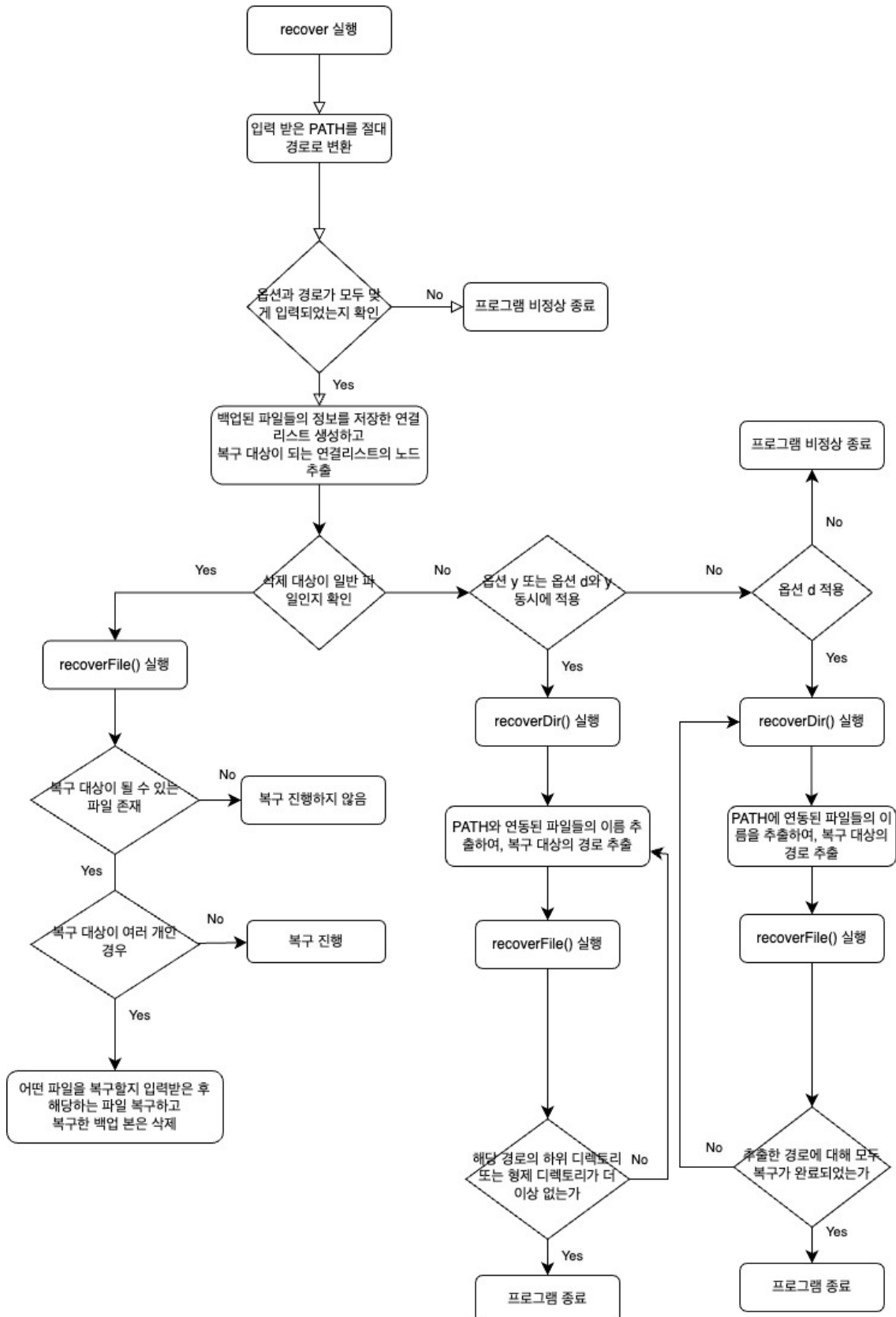
⑩ Remove 명령어

명령어 구현	-d 옵션	-r 옵션	-a 옵션
○	○	○	○



⑩ Recover 명령어

명령어 구현	-d 옵션	-r 옵션	-l 옵션	-n 옵션
○	○	○	○	×



#### ⑩ List 명령어

명령어 구현	rm 입력	rc 입력	vi(m) 입력	exit 입력
×	×	×	×	×

#### ⑩ Help 명령어

명령어 구현	COMMAND 입력 시 처리 구현
○	○

Help 명령어는 main.c에서 뿐만 아니라, 모든 내장 명령어에서 사용할 수 있도록 실행 함수를 따로 작성해주었다.

```
// command.c
void help(char *cmd)
{
    pid_t pid;
    int status;

    // 프로세스 생성 실패
    if((pid = fork()) < 0){
        fprintf(stderr, "fork error for help\n");
        exit(1);
    }

    // 자식 프로세스: help 프로그램 실행
    else if(pid == 0){
        execl("help", "help", cmd, NULL);
        exit(0);
    }

    // 부모 프로세스
    else{
        wait(&status);

        // 자식 프로세스가 정상적으로 종료되었으면 true 리턴
        // 만약 자식 프로세스가 비정상적으로 종료되었다면 부모 프로세스도 비정상 종료 처리
        if(!WIFEXITED(status))
            exit(1);
    }
}
```

#### 4. 실행 결과

##### 1) ssu\_backup

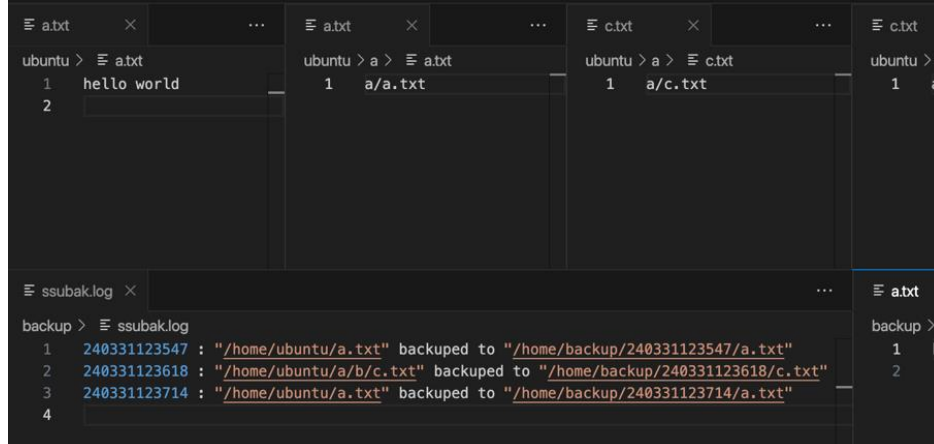
- ⑩ 프로그램 실행 시 /home 디렉토리 내에 /backup 디렉토리가 없다면 생성
- ⑩ 백업 디렉토리 내에 로그 파일이 없다면 생성

```
ubuntu@ip-172-31-41-81:~$ ls
assign assignment1 home task
ubuntu@ip-172-31-41-81:~$ cd ../
ubuntu@ip-172-31-41-81:/home$ ls
ubuntu
ubuntu@ip-172-31-41-81:/home$ cd ubuntu/assign
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help
Usage:
> backup <PATH> [OPTION]... : backup file if <PATH> is file
  -d: backup files in directory if <PATH> is directory
  -r: backup files in directory recursive if <PATH> is directory
  -y: backup file although already backedup
> remove <PATH> [OPTION]... : remove backedup file if <PATH> is file
  -d : remove backedup files in directory if <PATH> is directory
  -r : remove backedup files in directory recursive if <PATH> is directory
  -a : remove all backedup files
> recover <PATH> [OPTION]... : recover backedup file if <PATH> is file
  -d : recover backedup files in directory if <PATH> is directory
  -r : recover backedup files in directory recursive if <PATH> is directory
  -l : recover latest backedup file
  -n <NEW_PATH> : recover backedup file with new path
> list [PATH] : show backup list by directory structure
>> rm <INDEX> [OPTION]... : remove backedup files of [INDEX] with [OPTION]
>> rc <INDEX> [OPTION]... : recover backedup files of [INDEX] with [OPTION]
>> vi(m) <INDEX> : edit original file of [INDEX]
>> exit : exit program
> help [COMMAND] : show commands for program
ubuntu@ip-172-31-41-81:~/assign$ cd ../../
ubuntu@ip-172-31-41-81:/home$ ls
backup ubuntu
ubuntu@ip-172-31-41-81:/home$ cd backup/
ubuntu@ip-172-31-41-81:/home/backup$ ls
ssubak.log
```

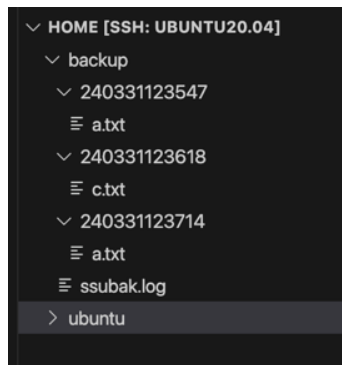
##### 2) 내장명령어 backup

- ⑩ 옵션이 없는 경우

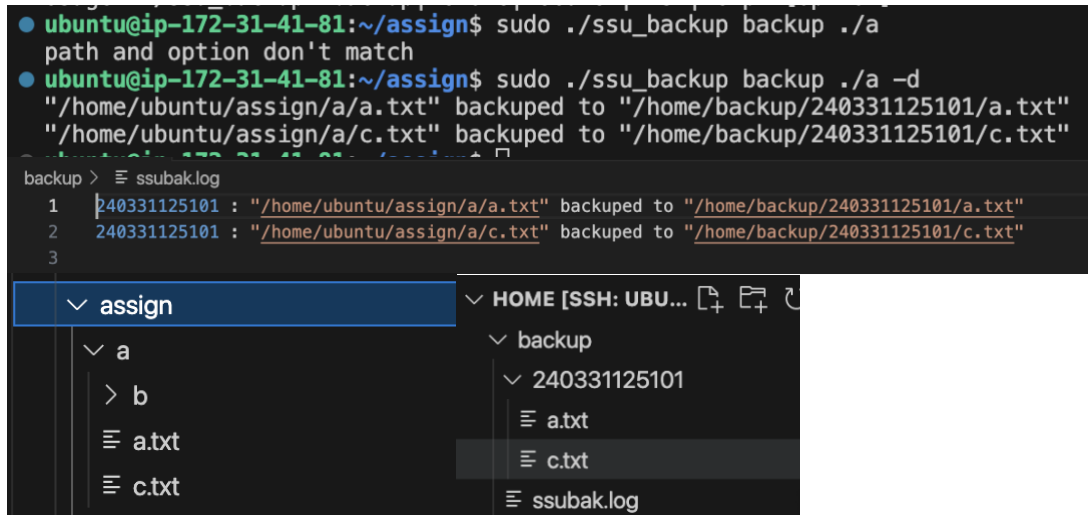
```
ubuntu@ip-172-31-41-81:/$ cd home/ubuntu/assign
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ../a.txt
"/home/ubuntu/a.txt" backedup to "/home/backup/240331123547/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ../a/b/c.txt
"/home/ubuntu/a/b/c.txt" backedup to "/home/backup/240331123618/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup /home/ubuntu/a.txt
"/home/ubuntu/a.txt" already backedup to "/home/backup/240331123547/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ echo "hello world" > ../a.txt
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ../a.txt
"/home/ubuntu/a.txt" backedup to "/home/backup/240331123714/a.txt"
```



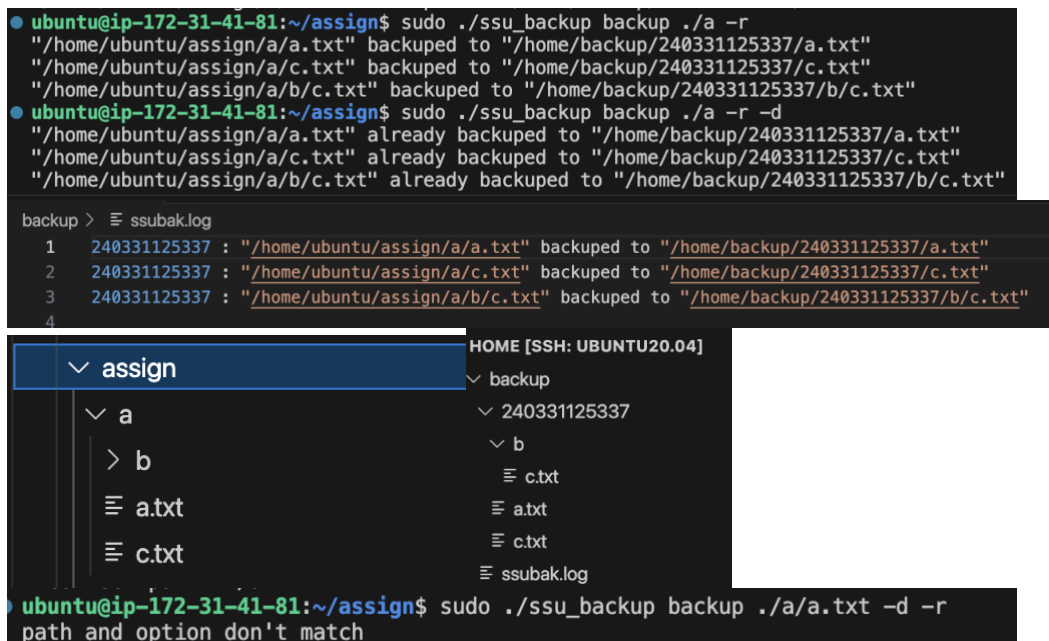
The screenshot shows four terminal windows. The first three windows show the output of the 'ssu\_backup' script for files 'a.txt', 'a/a.txt', and 'a/c.txt'. The fourth window shows the content of 'a.txt' as 'hello world'. Below these, a larger terminal window shows the output of the 'ssu\_backup' script for the file 'a.txt' again, showing it was already backed up. The output shows the file was backed up to '/home/backup/240331123547/a.txt' and then to '/home/backup/240331123714/a.txt'.



⑩ -d 옵션



⑩ -r 옵션



⑩ -y 옵션

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a -r
"/home/ubuntu/assign/a/a.txt" already backedup to "/home/backup/240331125337/a.txt"
"/home/ubuntu/assign/a/c.txt" already backedup to "/home/backup/240331125337/c.txt"
"/home/ubuntu/assign/a/b/c.txt" already backedup to "/home/backup/240331125337/b/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a -r -y
"/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331125757/a.txt"
"/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331125757/c.txt"
"/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331125757/b/c.txt"

backup > ≡ ssbak.log
1 240331125337 : "/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331125337/a.txt"
2 240331125337 : "/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331125337/c.txt"
3 240331125337 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331125337/b/c.txt"
4 240331125757 : "/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331125757/a.txt"
5 240331125757 : "/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331125757/c.txt"
6 240331125757 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331125757/b/c.txt"
7

```

### 3) 내장명령어 remove

#### ⑩ 옵션이 없는 경우

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a.txt
"/home/ubuntu/assign/a.txt" backedup to "/home/backup/240331130247/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ echo "hello world" > a.txt
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a.txt
"/home/ubuntu/assign/a.txt" backedup to "/home/backup/240331130347/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove ./a.txt
backup files of "/a.txt"
0. exit
1. 240331130247      5bytes
2. 240331130347     12bytes
>> 0
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove ./a.txt
backup files of "/a.txt"
0. exit
1. 240331130247      5bytes
2. 240331130347     12bytes
>> 1
"/home/backup/240331130247/a.txt" removed by "/home/ubuntu/assign/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove ./a.txt
"/home/backup/240331130347/a.txt" removed by "/home/ubuntu/assign/a.txt"

backup > ≡ ssbak.log
1 240331130247 : "/home/ubuntu/assign/a.txt" backedup to "/home/backup/240331130247/a.txt"
2 240331130347 : "/home/ubuntu/assign/a.txt" backedup to "/home/backup/240331130347/a.txt"
3 240331130406 : "/home/backup/240331130247/a.txt" removed by "/home/ubuntu/assign/a.txt"
4 240331130416 : "/home/backup/240331130347/a.txt" removed by "/home/ubuntu/assign/a.txt"
5

```

#### ⑩ -d 옵션

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a -r
"/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331130632/a.txt"
"/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331130632/c.txt"
"/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331130632/b/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a/b -d
"/home/ubuntu/assign/a/b/c.txt" already backedup to "/home/backup/240331130632/b/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a/b -d -y
"/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331130717/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove /home/ubuntu/assign/a -d
"/home/backup/240331130632/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
"/home/backup/240331130632/c.txt" removed by "/home/ubuntu/assign/a/c.txt"

backup > ≡ ssbak.log
1 240331130632 : "/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331130632/a.txt"
2 240331130632 : "/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331130632/c.txt"
3 240331130632 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331130632/b/c.txt"
4 240331130717 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331130717/c.txt"
5 240331130816 : "/home/backup/240331130632/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
6 240331130816 : "/home/backup/240331130632/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
7

```

#### ⑩ -y 옵션



```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove ./a -r -d
backup files of "/home/ubuntu/assign/a/a.txt"
0. exit
1. 240331131128      bytes
2. 240331131135      bytes
3. 240331131137      bytes
>> 0
backup files of "/home/ubuntu/assign/a/c.txt"
0. exit
1. 240331131128      bytes
2. 240331131135      bytes
3. 240331131137      bytes
>> 3
"/home/backup/240331131137/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
backup files of "/home/ubuntu/assign/a/b/c.txt"
0. exit
1. 240331131128      bytes
2. 240331131135      bytes
3. 240331131137      bytes
>> 3
"/home/backup/240331131137/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"

backup > ≡ ssubak.log
1 240331131128 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131128/a.txt"
2 240331131128 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131128/c.txt"
3 240331131128 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131128/b/c.txt"
4 240331131135 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131135/a.txt"
5 240331131135 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131135/c.txt"
6 240331131135 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131135/b/c.txt"
7 240331131137 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131137/a.txt"
8 240331131137 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131137/c.txt"
9 240331131137 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131137/b/c.txt"
10 240331131204 : "/home/backup/240331131137/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
11 240331131204 : "/home/backup/240331131137/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"
12

```

#### ⑩ -a 옵션

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup remove ./a -r -a
"/home/backup/240331131128/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
"/home/backup/240331131135/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
"/home/backup/240331131137/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
"/home/backup/240331131128/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
"/home/backup/240331131135/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
"/home/backup/240331131128/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"
"/home/backup/240331131135/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"

backup > ≡ ssubak.log
1 240331131128 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131128/a.txt"
2 240331131128 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131128/c.txt"
3 240331131128 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131128/b/c.txt"
4 240331131135 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131135/a.txt"
5 240331131135 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131135/c.txt"
6 240331131135 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131135/b/c.txt"
7 240331131137 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331131137/a.txt"
8 240331131137 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331131137/c.txt"
9 240331131137 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331131137/b/c.txt"
10 240331131204 : "/home/backup/240331131137/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
11 240331131204 : "/home/backup/240331131137/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"
12 240331131343 : "/home/backup/240331131128/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
13 240331131343 : "/home/backup/240331131135/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
14 240331131343 : "/home/backup/240331131137/a.txt" removed by "/home/ubuntu/assign/a/a.txt"
15 240331131343 : "/home/backup/240331131128/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
16 240331131343 : "/home/backup/240331131135/c.txt" removed by "/home/ubuntu/assign/a/c.txt"
17 240331131343 : "/home/backup/240331131128/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"
18 240331131343 : "/home/backup/240331131135/b/c.txt" removed by "/home/ubuntu/assign/a/b/c.txt"
19

```

#### 4) 내장명령어 recover

##### ⑩ 옵션이 없는 경우

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a.txt
"/home/ubuntu/assign/a.txt" backed up to "/home/backup/240331132036/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a/a.txt
"/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331132038/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a.txt
"/home/backup/240331132036/a.txt" is not changed with "/home/ubuntu/assign/a.txt"
ubuntu@ip-172-31-41-81:~/assign$ echo "hello world" > ./a.txt
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a.txt
"/home/backup/240331132036/a.txt" recovered to "/home/ubuntu/assign/a.txt"

```

HOME [SSH: UBU...]

backup > ssubak.log

- 1 240331132036 : "/home/ubuntu/assign/a.txt" backed up to "/home/backup/240331132036/a.txt"
- 2 240331132038 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331132038/a.txt"
- 3 240331132059 : "/home/backup/240331132036/a.txt" recovered to "/home/ubuntu/assign/a.txt"
- 4

backup

- 240331132038
- a.txt
- ssubak.log

ubuntu

# ⑩ -d 옵션

```

ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a -d
backup files of "/home/ubuntu/assign/a/a.txt"
0. exit
1. 240331132458      13bytes
2. 240331132501      13bytes
>> 1
"/home/backup/240331132458/a.txt" is not changed with "/home/ubuntu/assign/a/a.txt"
backup files of "/home/ubuntu/assign/a/c.txt"
0. exit
1. 240331132458      7bytes
2. 240331132501      7bytes
>> 2
"/home/backup/240331132501/c.txt" is not changed with "/home/ubuntu/assign/a/c.txt"
ubuntu@ip-172-31-41-81:~/assign$ echo "hello" > ./a/a.txt
ubuntu@ip-172-31-41-81:~/assign$ echo "hello" > ./a/c.txt
ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a -d
backup files of "/home/ubuntu/assign/a/a.txt"
0. exit
1. 240331132458      13bytes
2. 240331132501      13bytes
>> 1
"/home/backup/240331132458/a.txt" recovered to "/home/ubuntu/assign/a/a.txt"
backup files of "/home/ubuntu/assign/a/c.txt"
0. exit
1. 240331132458      7bytes
2. 240331132501      7bytes
>> 2
"/home/backup/240331132501/c.txt" recovered to "/home/ubuntu/assign/a/c.txt"

```

backup > ssubak.log

- 1 240331132458 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331132458/a.txt"
- 2 240331132458 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331132458/c.txt"
- 3 240331132458 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331132458/b/c.txt"
- 4 240331132501 : "/home/ubuntu/assign/a/a.txt" backed up to "/home/backup/240331132501/a.txt"
- 5 240331132501 : "/home/ubuntu/assign/a/c.txt" backed up to "/home/backup/240331132501/c.txt"
- 6 240331132501 : "/home/ubuntu/assign/a/b/c.txt" backed up to "/home/backup/240331132501/b/c.txt"
- 7 240331132603 : "/home/backup/240331132458/a.txt" recovered to "/home/ubuntu/assign/a/a.txt"
- 8 240331132603 : "/home/backup/240331132501/c.txt" recovered to "/home/ubuntu/assign/a/c.txt"
- 9

# ⑩ -r 옵션



```

/home/backup/240331133441/c.txt" recoverd to "/home/ubuntu/assign/a/c.txt"
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a
path and option don't match
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a/a.txt -d
path and option don't match
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a/a.txt -d -r
path and option don't match
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a -r
backup files of "/home/ubuntu/assign/a/b/a.txt"
0. exit
1. 240331133441          4bytes
2. 240331133448          4bytes
>> 1
"/home/backup/240331133441/b/a.txt" recoverd to "/home/ubuntu/assign/a/b/a.txt"
backup files of "/home/ubuntu/assign/a/b/c.txt"
0. exit
1. 240331133441          3bytes
2. 240331133448          3bytes
>> 1
"/home/backup/240331133441/b/c.txt" recoverd to "/home/ubuntu/assign/a/b/c.txt"
p > ≡ ssubak.log
240331133441 : "/home/ubuntu/assign/a/b/a.txt" backedup to "/home/backup/240331133441/b/a.txt"
240331133441 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331133441/b/c.txt"
240331133448 : "/home/ubuntu/assign/a/b/a.txt" backedup to "/home/backup/240331133448/b/a.txt"
240331133448 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331133448/b/c.txt"
240331133549 : "/home/backup/240331133441/b/a.txt" recoverd to "/home/ubuntu/assign/a/b/a.txt"
240331133549 : "/home/backup/240331133441/b/c.txt" recoverd to "/home/ubuntu/assign/a/b/c.txt"

```

#### ⑩ -i 옵션

```

● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a -r
"/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331132941/a.txt"
"/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331132941/c.txt"
"/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331132941/b/c.txt"
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup backup ./a -r -y
"/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331132949/a.txt"
"/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331132949/c.txt"
"/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331132949/b/c.txt"
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup recover ./a -r -l
"/home/backup/240331132949/a.txt" recoverd to "/home/ubuntu/assign/a/a.txt"
"/home/backup/240331132949/c.txt" recoverd to "/home/ubuntu/assign/a/c.txt"
"/home/backup/240331132949/b/c.txt" recoverd to "/home/ubuntu/assign/a/b/c.txt"
backup > ≡ ssubak.log
1 240331132941 : "/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331132941/a.txt"
2 240331132941 : "/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331132941/c.txt"
3 240331132941 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331132941/b/c.txt"
4 240331132949 : "/home/ubuntu/assign/a/a.txt" backedup to "/home/backup/240331132949/a.txt"
5 240331132949 : "/home/ubuntu/assign/a/c.txt" backedup to "/home/backup/240331132949/c.txt"
6 240331132949 : "/home/ubuntu/assign/a/b/c.txt" backedup to "/home/backup/240331132949/b/c.txt"
7 240331133036 : "/home/backup/240331132949/a.txt" recoverd to "/home/ubuntu/assign/a/a.txt"
8 240331133036 : "/home/backup/240331132949/c.txt" recoverd to "/home/ubuntu/assign/a/c.txt"
9 240331133036 : "/home/backup/240331132949/b/c.txt" recoverd to "/home/ubuntu/assign/a/b/c.txt"
10

```

#### 5) 내장명령어 help

```

● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help
Usage:
> backup <PATH> [OPTION]... : backup file if <PATH> is file
  -d: backup files in directory if <PATH> is directory
  -r: backup files in directory recursive if <PATH> is directory
  -y: backup file although already backedup
> remove <PATH> [OPTION]... : remove backedup file if <PATH> is file
  -d: remove backedup files in directory if <PATH> is directory
  -r: remove backedup files in directory recursive if <PATH> is directory
  -a: remove all backedup files
> recover <PATH> [OPTION]... : recover backedup file if <PATH> is file
  -d: recover backedup files in directory if <PATH> is directory
  -r: recover backedup files in directory recursive if <PATH> is directory
  -l: recover latest backedup file
  -n <NEW_PATH> : recover backedup file with new path
> list [PATH] : show backup list by directory structure
  >> rm <INDEX> [OPTION]... : remove backedup files of [INDEX] with [OPTION]
  >> rc <INDEX> [OPTION]... : recover backedup files of [INDEX] with [OPTION]
  >> vi(m) <INDEX> : edit original file of [INDEX]
  >> exit : exit program
> help [COMMAND] : show commands for program
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help backup
Usage: backup <PATH> [OPTION]... : backup file if <PATH> is file
  -d: backup files in directory if <PATH> is directory
  -r: backup files in directory recursive if <PATH> is directory
  -y: backup file although already backedup
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help remove
Usage: remove <PATH> [OPTION]... : remove backedup file if <PATH> is file
  -d: remove backedup files in directory if <PATH> is directory
  -r: remove backedup files in directory recursive if <PATH> is directory
  -a: remove all backedup files
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help recover
Usage: recover <PATH> [OPTION]... : recover backedup file if <PATH> is file
  -d: recover backedup files in directory if <PATH> is directory
  -r: recover backedup files in directory recursive if <PATH> is directory
  -l: recover latest backedup file
  -n <NEW_PATH> : recover backedup file with new path
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help list
Usage: list [PATH] : show backup list by directory structure
  >> rm <INDEX> [OPTION]... : remove backedup files of [INDEX] with [OPTION]
  >> rc <INDEX> [OPTION]... : recover backedup files of [INDEX] with [OPTION]
  >> vi(m) <INDEX> : edit original file of [INDEX]
  >> exit : exit program
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help help
Usage: help [COMMAND] : show commands for program
● ubuntu@ip-172-31-41-81:~/assign$ sudo ./ssu_backup help aaa
Usage: help [backup|remove|recover|list|help]

```