

1. 과제 개요

특정 디렉토리 아래의 파일 변경 사항을 감지하는 데몬 프로세스를 관리한다. 또한, 특정 디렉토리의 구조를 트리 형태로 보여주는 기능으로 구성되어 있다. 즉, 과제의 주 기능은 디렉터리를 트리 형태로 관리하거나 최소한 트리의 관점에서 바라보는 방향에서 구현되었다.

2. 기능

2-1. 데몬 프로세스 생성 (add 명령어)

내부 명령어 중 하나인 add는 데몬 프로세스를 생성한다. 생성에 성공하면 monitor_list.txt에 디렉터리 경로와 데몬 프로세스 아이디(이하 pid)를 줄 단위로 작성한다. 만약, 이미 다른 데몬 프로세스에 의해 모니터링 되는 디렉토리의 경로와 포함 관계(포함하거나, 포함되거나)에 있다면 데몬 프로세스는 새롭게 생성되지 않는다. 또한, 기본 1초인 데몬 프로세스의 모니터링 주기를 -t 인자를 사용하여 변경할 수 있다.

2-2. 데몬 프로세스 종료 (delete 명령어)

monitor_list.txt에 존재하는 pid를 입력받으면 SIGUSR1 시그널을 데몬 프로세스에 전송하여 프로세스를 종료 시킨다.

2-3. 디렉터리 구조 출력 (tree 명령어)

monitor_list.txt에 존재하는 경로를 입력받으면 기본 tree 명령어와 유사하게 디렉터리의 구조를 출력한다.

2-4. 도움말 출력 (help 명령어)

프로그램의 모든 내부 명령어의 도움말을 출력한다.

2-5. 프로그램 종료 (exit 명령어)

프로그램을 종료한다. 데몬 프로세스는 종료되지 않는다.

2-6. 파일 변경 사항 감지 (데몬 프로세스)

데몬 프로세스는 이전 상태에서의 디렉터리 구조를 트리 형태로 가지고 있다. 그리고 현재 디렉터리 구조를 다시 트리 형태로 가져온다. 그리고 간단한 재귀 및 투 포인터 알고리즘을 사용하여 두 트리 사이의 차이점을 비교한다.

3. 설계

프로그램의 진입점은 ssu_monitor.c이다. 각각 내부 명령어는 commands/*.c의 파일에 의해 생성되는 각각의 실행 파일로 다시 실행된다. 또한, 데몬 프로세스는 daemon/daemon_process.c에 의해 생성되는 실행 파일로 실행된다. 그 외에 commons/는 공통으로 사용되는 함수의 모듈, tree/는 tree 명령어, daemon/은 데몬 프로세스에서 사용되는 코드가 구현되어 있다.

3-1. 파일(모듈) 전체 목록

프로젝트의 파일(모듈) 구성은 하기와 같다.

```
commands/
    command_add.c
    command_delete.c
    command_help.c
    command_tree.c
commons/
    const.h
    daemon_manager.{c, h}
    path.{c, h}
    usage.{c, h}
daemon/
    daemon_process.c
    daemon_tree.{c, h}
tree/
    tree.{c, h}
Makefile
ssu_monitor.c
```

3-2. 소스코드별 함수 개요

순서도에 표시되는 주요 함수만 작성하였다.

3-2-1. ssu_monitor.c

메인 프로그램의 실행 파일이다.

3-2-1-1. int main(int argc, char *argv[])

내장 명령어를 실행시키기 위한 간단한 PATH 환경변수 추가 작업, monitor_list.txt 경로 작업을 진행한다. 그런 후, 프롬프트를 작동시켜 한 줄마다 입력받아 공백 단위로 구분하여 내부 명령어 프로세스를 실행하거나 프로세스 종료 작업 등을 진행한다.

3-2-2. commands/command_add.c

내부 명령어 add의 실행 파일이다.

3-2-2-1. int main(int argc, char *argv[])

내부 명령어 add의 진입점이다. getopt와 argc/argv 등을 사용하여 명령어 인자를 처리한다. 그리고, daemon_manager.c의 add_daemon_process 함수를 호출하여 데몬 프로세스를 생성한다.

3-2-3. commands/command_delete.c

내부 명령어 delete의 실행 파일이다.

3-2-3-1. int main(int argc, char *argv[])

내부 명령어 delete의 진입점이다. 명령어 인자를 처리한 후, daemon_manager.c의 delete_daemon_process 함수를 호출하여 데몬 프로세스를 종료한다.

3-2-4. commands/command_help.c

내부 명령어 help의 실행 파일이다.

3-2-4-1. int main(int argc, char *argv[])

내부 명령어 help의 진입점이다. 프로그램의 모든 내부 명령어를 출력한다.

3-2-5. commands/command_tree.c

내부 명령어 tree의 실행 파일이다.

3-2-5-1. int main(int argc, char *argv[])

내부 명령어 tree의 진입점이다. path가 monitor_list.txt에 존재하면 print_as_tree 함수를 호출하여 해당 경로의 디렉터리 구조를 트리 형태로 출력한다.

3-2-6. commons/const.h

프로그램 전역에서 사용되는 매크로 상수(BUFFER_MAX)가 정의되어 있다.

3-2-7. commons/daemon_manager.{c, h}

데몬 프로세스와 monitor_list.txt를 관리하는 코드이다.

3-2-7-1. int can_add_path(const char *path)

monitor_list.txt로부터 이미 모니터링 중인 경로와 포함 관계에 있으면 0을, 그렇지 않으면 1을 리턴한다.

3-2-7-2. char *exist_pid(pid_t pid, char *resolved)

monitor_list.txt로부터 pid가 등록된 데몬 프로세스의 pid이면 모니터링 중인 경로를 resolved에 복사한 후 resolved의 경로를, 그렇지 않으면 NULL을 리턴한다.

3-2-7-3. void append_path(const char *path, pid_t pid)

monitor_list.txt의 마지막 줄에 path와 pid를 추가한다.

3-2-7-4. void remove_pid(pid_t pid)

monitor_list.txt에서 pid로 등록된 데몬 프로세스의 정보를 삭제한다.

3-2-7-5. int add_daemon_process(const char *path, int repeat_time)

데몬 프로세스를 추가하는 함수이다. 데몬 프로세스를 생성하며, append_path 함수를 사용하여 파일에 정보를 저장한다.

3-2-7-6. char *delete_daemon_process(pid_t pid, char *resolved)

exist_pid를 먼저 호출하여 해당 pid가 모니터링 중인 폴더의 경로를 먼저 resolved에 저장한다. 그런 후, SIGUSR1 신호를 보내 프로세스를 종료시킨다.

3-2-7-7. int exist_path_exactly(const char *path)

해당 경로와 정확히 일치하는 데몬 프로세스가 존재한다면 1을, 그렇지 않으면 0을 리턴한다.

3-2-7-8. void init_filesystem()

monitor_list.txt의 경로 저장, 파일 생성 등의 작업을 진행한다.

3-2-8. commons/path.{c, h}

경로와 관련된 작업을 처리한다.

3-2-8-1. void set_additional_path(const char *executable_path)

현재 작업 디렉터리와 실행 파일의 디렉터리를 환경변수 PATH에 추가적으로 등록하는 함수이다.

3-2-9. usage/usage.{c, h}

3-2-9-1. void print_usage()

프로그램의 전체 도움말을 출력한다.

3-2-9-2. void print_usage_add()

add 명령어의 도움말을 출력한다.

3-2-9-3. void print_usage_delete()

delete 명령어의 도움말을 출력한다.

3-2-9-4. void print_usage_tree()

tree 명령어의 도움말을 출력한다.

3-2-10. daemon/daemon_process.c

데몬 프로세스의 실행 파일이다.

3-2-10-1. int main(int argc, char *argv[])

데몬 프로세스의 진입점이다. 모니터링할 경로와 반복 시간을 argv로 받은 후, daemon_tree에 구현된 함수를 사용하여 모니터링을 진행한다. 또한, SIGUSR1 시그널 핸들러를 등록한다.

3-2-10-2. void stop_by_signal(int sig)

SIGUSR1 시그널이 들어오면 log.txt에 exit을 출력한 후 프로그램을 종료한다.

3-2-11. daemon/daemon_tree.{c, h}

3-2-11-1. void log_file(FILE *fp, const char *head, const char *body)

해당 파일에 로그를 출력한다. [DATE][<head>][<body>] 형식으로 출력된다.

3-2-11-2. int cmp_by_filename_asc(const void *p1, const void *p2)

DaemonFileNode를 파일 경로 순으로 정렬한다.

3-2-11-3. struct DaemonFileNode *construct_tree(const char *path)

path를 루트로 하는 서브 트리를 생성한다. path가 일반 파일이면 is_dir=0을 넣고 노드를 바로 리턴한다.

path가 디렉터리라면 is_dir=1로 지정되고, 재귀적으로 construct_tree를 호출하며 트리를 완성한다.

3-2-11-4. void free_tree(struct DaemonFileNode *node)

트리를 재귀적으로 순회하면서 할당된 모든 동적 메모리를 해제한다.

3-2-11-5. void compare_tree(struct DaemonFileNode *prev, struct DaemonFileNode *next, FILE *fp, const char *logPath)

이전 트리와 다음 트리를 비교하여 변경점을 파일에 작성한다. 이 때, 로그 파일은 변경되더라도 변경점을 작성하지 않는다. 현재 노드가 디렉터리라면 하위 파일에 대해서도 재귀적으로 compare_tree를 호출한다. 이 때, 하위 파일들은 앞서 cmp_by_filename_asc 함수로 인해 정렬된 상태이므로 투 포인터(two-pointer) 알고리즘을 사용하여 prev와 next에서 이름이 같은 하위 파일들을 선형 시간에 찾아 동작한다.

3-2-12. tree/tree.{c, h}

3-2-12-1. const char *get_filename(const char *path)

경로로부터 파일 이름을 가져온다.

3-2-12-2. int cmp_by_filename_asc(const void *p1, const void *p2)

struct dirent를 파일 경로 순으로 정렬한다.

3-2-12-3. void traverse(const char *path)

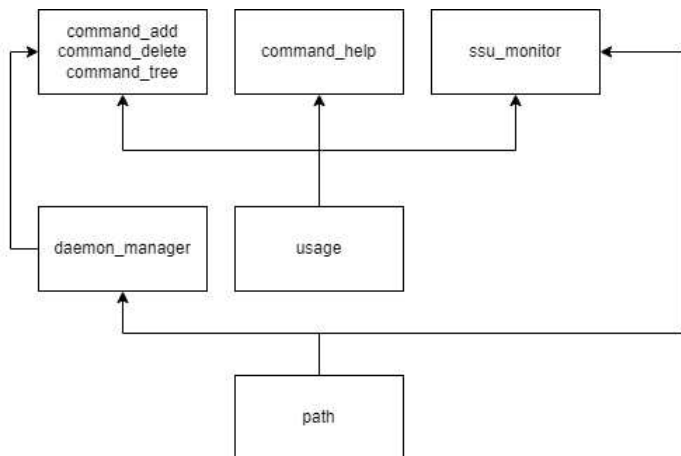
트리 형태로 출력을 진행한다. 현재 노드까지 오기 전의 조상(ancestor) 노드들이 마지막 자식에 도달했는지, 도달하지 않았는지, 현재 노드가 디렉터리인지 아닌지 등에 따라 출력 포맷을 잘 정하여 출력한다.

3-2-12-4. void print_as_tree(const char *path)

path를 트리 형태로 출력한다.

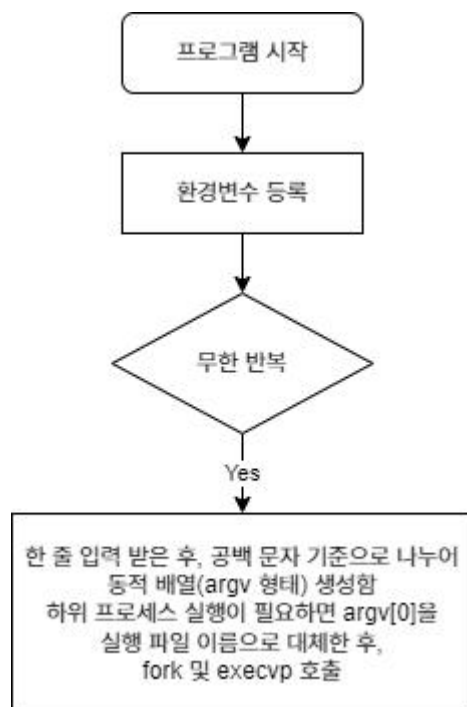
3-3. 종속성(dependency) 그래프

각 모듈(source, header file)이 어떤 모듈에 의존하는지 나타냈다. 작성된 Makefile의 종속성도 아래와 같으므로 Makefile의 종속성에 대한 내용은 생략한다.

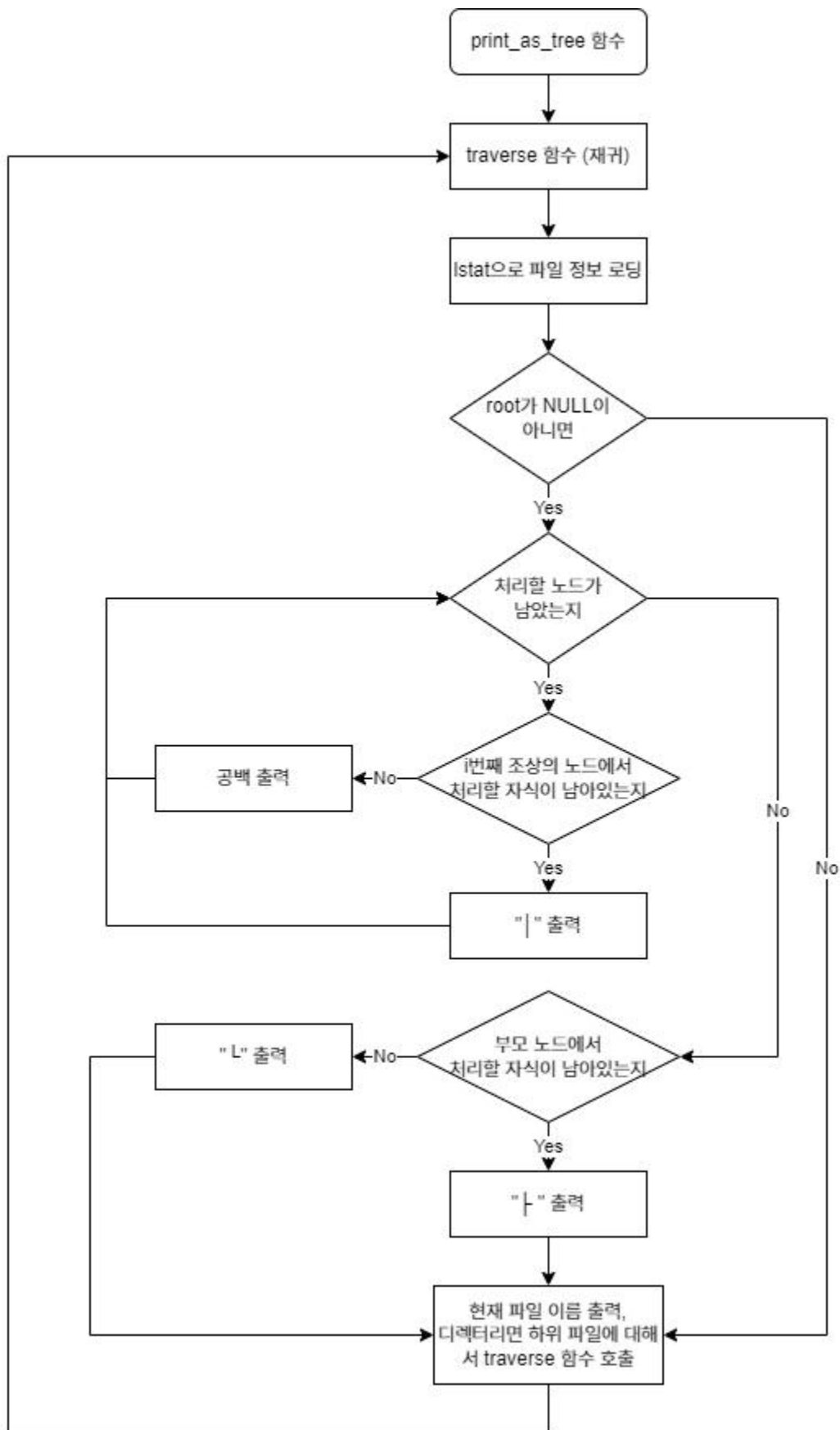


3-4. 순서도

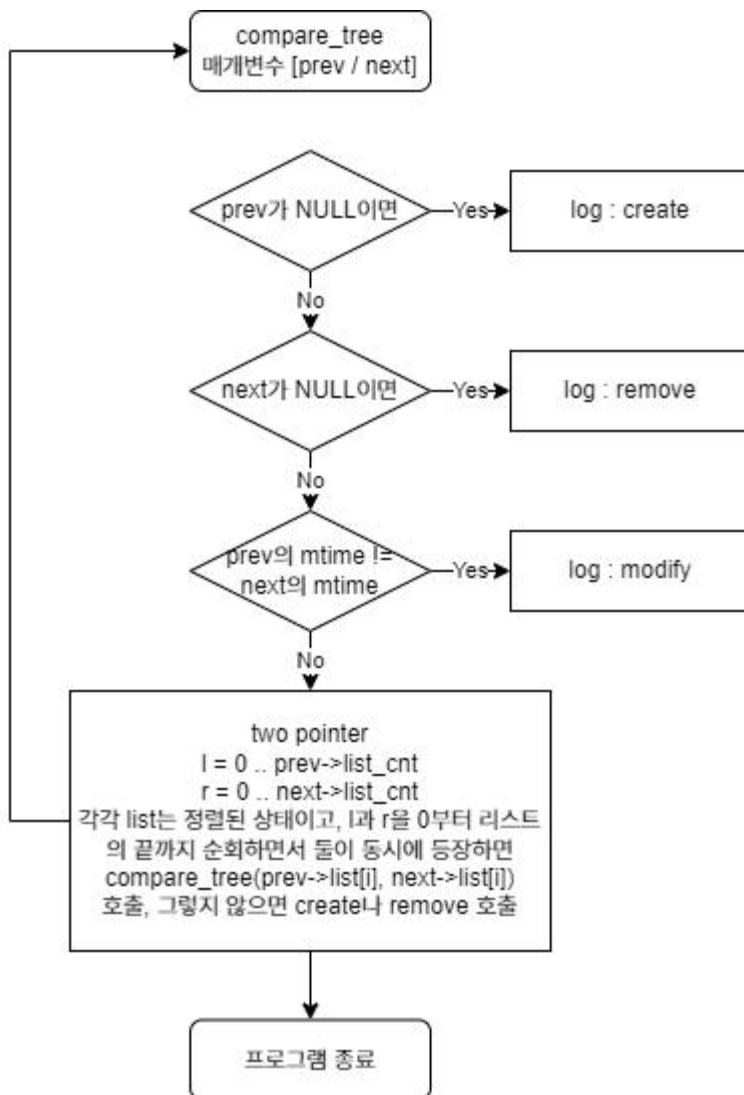
3-4-1. 메인 프로그램 순서도



3-4-2. tree 명령어 순서도



3-4-3. compare_tree [데몬 프로세스 트리 비교]



4. 실행결과

4-1. 메인 프로그램 (ssu_monitor)

4-1-1. 프롬프트 동작 확인

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494>
20221494> help
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
> delete <DAEMON_PID>
> tree <DIRPATH>
> help
> exit
20221494>
```

엔터만 입력 시 프롬프트 재출력 등

4-1-2. [예외 처리] 지정된 명령어 이외 입력

```
20221494> asdf
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
> delete <DAEMON_PID>
> tree <DIRPATH>
> help
> exit
20221494>
```

4-2. 내장 명령어 add

4-2-1. 상대 경로 입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> |
```

4-2-2. 절대 경로 입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add /home/nlog/p3
monitoring started (/home/nlog/p3)
20221494>
```

4-2-3. -t 인자 입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add . -t 10
monitoring started (/home/nlog/p3)
20221494> |
```


4-2-4. monitor_list.txt 확인

```
nlog@nlog-vm-lsp:~/p3$ cat monitor_list.txt
/home/nlog/p3 6319
nlog@nlog-vm-lsp:~/p3$ |
```

4-2-5. [예외 처리] 포함 관계에 속하는 경로 입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add . -t 10
monitoring started (/home/nlog/p3)
20221494> add .
monitoring failed (/home/nlog/p3)
20221494> add ..
monitoring failed (/home/nlog)
20221494> add ./commands
monitoring failed (/home/nlog/p3/commands)
20221494> |
```

4-2-6. [예외 처리] 첫 번째 인자 예외 처리

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
20221494> add asdf
(asdf) doesn't exist.
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
20221494> add ssu_monitor
(ssu_monitor) is not directory.
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
20221494> |
```

인자 미입력 / 존재하지 않는 경로 /
디렉터리가 아닌 경로

4-3. 내장 명령어 delete

4-3-1. 삭제 확인

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> delete 6365
monitoring ended (/home/nlog/p3)
20221494>
```

추가 후 삭제

```
nlog@nlog-vm-lsp:~/p3$ cat monitor_list.txt
/home/nlog/p3 6365
nlog@nlog-vm-lsp:~/p3$ cat monitor_list.txt
nlog@nlog-vm-lsp:~/p3$
```

monitor_list.txt 확인

4-3-2. [예외 처리] pid 존재하지 않는 경우

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> delete 6381
monitoring deletion failed
20221494> delete 6382
monitoring ended (/home/nlog/p3)
20221494> |
```

추가 및 삭제

```
nlog@nlog-vm-lsp:~/p3$ cat monitor_list.txt
/home/nlog/p3 6382
nlog@nlog-vm-lsp:~/p3$ |
```

monitor_list.txt (pid 확인)

4-3-3. [예외 처리] pid 미입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> delete
Usage:
> delete <DAEMON_PID>
20221494>
```

4-4. 내장 명령어 tree

4-4-1. 동작 확인

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> tree .
p3
├── .clion.source.upload.marker
├── Makefile
├── command_add
├── command_delete
├── command_help
├── command_tree
├── commands
│   ├── command_add.c
│   ├── command_delete.c
│   ├── command_help.c
│   └── command_tree.c
├── commons
│   ├── const.h
│   ├── daemon_manager.c
│   ├── daemon_manager.h
│   ├── path.c
│   ├── path.h
│   ├── usage.c
│   └── usage.h
├── daemon
│   ├── daemon_process.c
│   ├── daemon_tree.c
│   └── daemon_tree.h
├── daemon_manager.o
├── daemon_process
├── daemon_tree.o
├── log.txt
├── monitor_list.txt
├── path.o
├── ssu_monitor
├── ssu_monitor.c
├── tree
│   ├── tree.c
│   └── tree.h
├── tree.o
└── usage.o
20221494>
```

상대 경로 입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> tree /home/nlog/p3
p3
├── .clion.source.upload.marker
├── Makefile
├── command_add
├── command_delete
├── command_help
├── command_tree
├── commands
│   ├── command_add.c
│   ├── command_delete.c
│   ├── command_help.c
│   └── command_tree.c
├── commons
│   ├── const.h
│   ├── daemon_manager.c
│   ├── daemon_manager.h
│   ├── path.c
│   ├── path.h
│   ├── usage.c
│   └── usage.h
├── daemon
│   ├── daemon_process.c
│   ├── daemon_tree.c
│   └── daemon_tree.h
├── daemon_manager.o
├── daemon_process
├── daemon_tree.o
├── log.txt
├── monitor_list.txt
├── path.o
├── ssu_monitor
├── ssu_monitor.c
├── tree
│   ├── tree.c
│   └── tree.h
├── tree.o
└── usage.o
20221494>
```

절대 경로 입력

4-4-2. [예외 처리] 존재하지 않는 경로

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> tree ..
(/home/nlog) doesn't exist in (/home/nlog/p3/monitor_list.txt).
20221494>
```

4-4-3. [예외 처리] 경로 미입력

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> tree
Usage:
> tree <DIRPATH>
20221494> |
```

4-5. 내장 명령어 help

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> help
Usage:
> add <DIRPATH> [OPTION]
-t <TIME> : set repeat time
> delete <DAEMON_PID>
> tree <DIRPATH>
> help
> exit
20221494> |
```

4-6. 내장 명령어 exit

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> exit
nlog@nlog-vm-lsp:~/p3$ |
```

4-7. 데몬 프로세스 동작 확인

- 초기 실행 및 메인 프로그램 종료

```
nlog@nlog-vm-lsp:~/p3$ ./ssu_monitor
20221494> add .
monitoring started (/home/nlog/p3)
20221494> exit
nlog@nlog-vm-lsp:~/p3$ cat monitor_list.txt
/home/nlog/p3 6443
nlog@nlog-vm-lsp:~/p3$ cat log.txt
nlog@nlog-vm-lsp:~/p3$ |
```

- 단순 파일 생성/삭제

```
nlog@nlog-vm-lsp:~/p3$ touch asdf
nlog@nlog-vm-lsp:~/p3$ touch ksalf
nlog@nlog-vm-lsp:~/p3$ rm asdf
nlog@nlog-vm-lsp:~/p3$ rm ksalf
nlog@nlog-vm-lsp:~/p3$
```

```
nlog@nlog-vm-lsp:~/p3$ tail -f log.txt
[2023-05-05 15:26:20][create][/home/nlog/p3/asdf]
[2023-05-05 15:26:23][create][/home/nlog/p3/ksalf]
[2023-05-05 15:26:25][remove][/home/nlog/p3/asdf]
[2023-05-05 15:26:26][remove][/home/nlog/p3/ksalf]
|
```

- 동시에 여러 파일 삭제 (rm -r 사용)

```
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/Makefile]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/command_add]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/command_delete]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/command_help]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/command_tree]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commands/command_add.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commands/command_delete.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commands/command_help.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commands/command_tree.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/const.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/daemon_manager.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/daemon_manager.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/path.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/path.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/usage.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/commons/usage.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon/daemon_process.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon/daemon_tree.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon/daemon_tree.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon_manager.o]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon_process]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/daemon_tree.o]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/log.txt]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/monitor_list.txt]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/path.o]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/ssu_monitor]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/ssu_monitor.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/tree/tree.c]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/tree/tree.h]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/tree.o]
[2023-05-14 15:29:10][remove][/home/nlog/Downloads/p3/usage.o]
```

- 동시에 여러 파일 생성 (cp 사용)

```
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/Makefile]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/command_add]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/command_delete]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/command_help]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/command_tree]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commands/command_add.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commands/command_delete.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commands/command_help.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commands/command_tree.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/const.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/daemon_manager.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/daemon_manager.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/path.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/path.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/usage.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/commons/usage.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon/daemon_process.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon/daemon_tree.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon/daemon_tree.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon_manager.o]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon_process]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/daemon_tree.o]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/log.txt]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/monitor_list.txt]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/path.o]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/ssu_monitor]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/ssu_monitor.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/tree/tree.c]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/tree/tree.h]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/tree.o]
[2023-05-14 15:28:44][create][/home/nlog/Downloads/p3/usage.o]
```

- 파일 수정 (touch 사용)

```
nlog@nlog-vm-lsp:~/p3$ touch ssu_monitor
nlog@nlog-vm-lsp:~/p3$ touch command*
nlog@nlog-vm-lsp:~/p3$
```

```
[2023-05-05 15:29:40][create][[/home/nlog/p3/commands]]
[2023-05-05 15:29:40][modify][[/home/nlog/p3/ssu_monitor]]
[2023-05-05 15:29:47][modify][[/home/nlog/p3/command_add]]
[2023-05-05 15:29:47][modify][[/home/nlog/p3/command_delete]]
[2023-05-05 15:29:47][modify][[/home/nlog/p3/command_help]]
[2023-05-05 15:29:47][modify][[/home/nlog/p3/command_tree]]
```

4-8. Makefile

4-8-1. build

```
nlog@nlog-vm-lsp:~/p3$ make
Build complete
nlog@nlog-vm-lsp:~/p3$
```

4-8-2. clean

```
nlog@nlog-vm-lsp:~/p3$ make clean
rm -f ssu_monitor daemon_process command_* *.o
nlog@nlog-vm-lsp:~/p3$
```