

Report

HW#1 파일 무결성 검사

고급 프로그래밍 실습

Dpt. of Mobile System Engineering
32192530 양윤성

2023.11.22.

● Contents

1. Abstract	1p
2. Introduction	2p
3. Design	3p
4. Implementation	5p
5. Test and Results	6p
6. Conclusion	14p

1. Abstract

본 프로그램은 파일들의 해시값을 이용하여 파일의 무결성을 검사하는 프로그램이다. 프로그램 이름은 `check_intgerity`이며, 파일의 무결성을 검사할 최상위 디렉터리의 경로를 매개변수로 받는다. 그럼 매개변수로 받은 경로의 하위 디렉터리에 있는 모든 파일들을 검사한다. 이 파일들의 해시값 정보는 `hash.txt`에 저장되어 있는데, `hash.txt`의 디렉터리는 `check_integrity.sh`와 동일한 디렉터리 내 `config`파일에 적혀 있다. 그래서 만약 `hash.txt`에 정보가 없는 파일을 발견했다면, 새로운 파일임을 알리고 해당 파일의 해시값을 기록하여 저장한다. 그러나 파일이 이미 존재하면 2가지 조건을 확인하게 된다. 해당 파일의 해시값이 변경되지 않았다면 무결성이 검증된 것이므로 통과하지만, 해시값이 변경되었다면 변경되었다는 메시지를 사용자에게 출력한다. 여기서 사용자의 선택에 따라 변경된 값으로 해시값을 대체할지 말지를 결정한다. 변경한다면 새로운 해시값으로 `hash.txt`에 저장하게 된다. 이 프로그램을 쉘 스크립트로 구현하는 것이 이번 과제의 목표이다.

2. Introduction

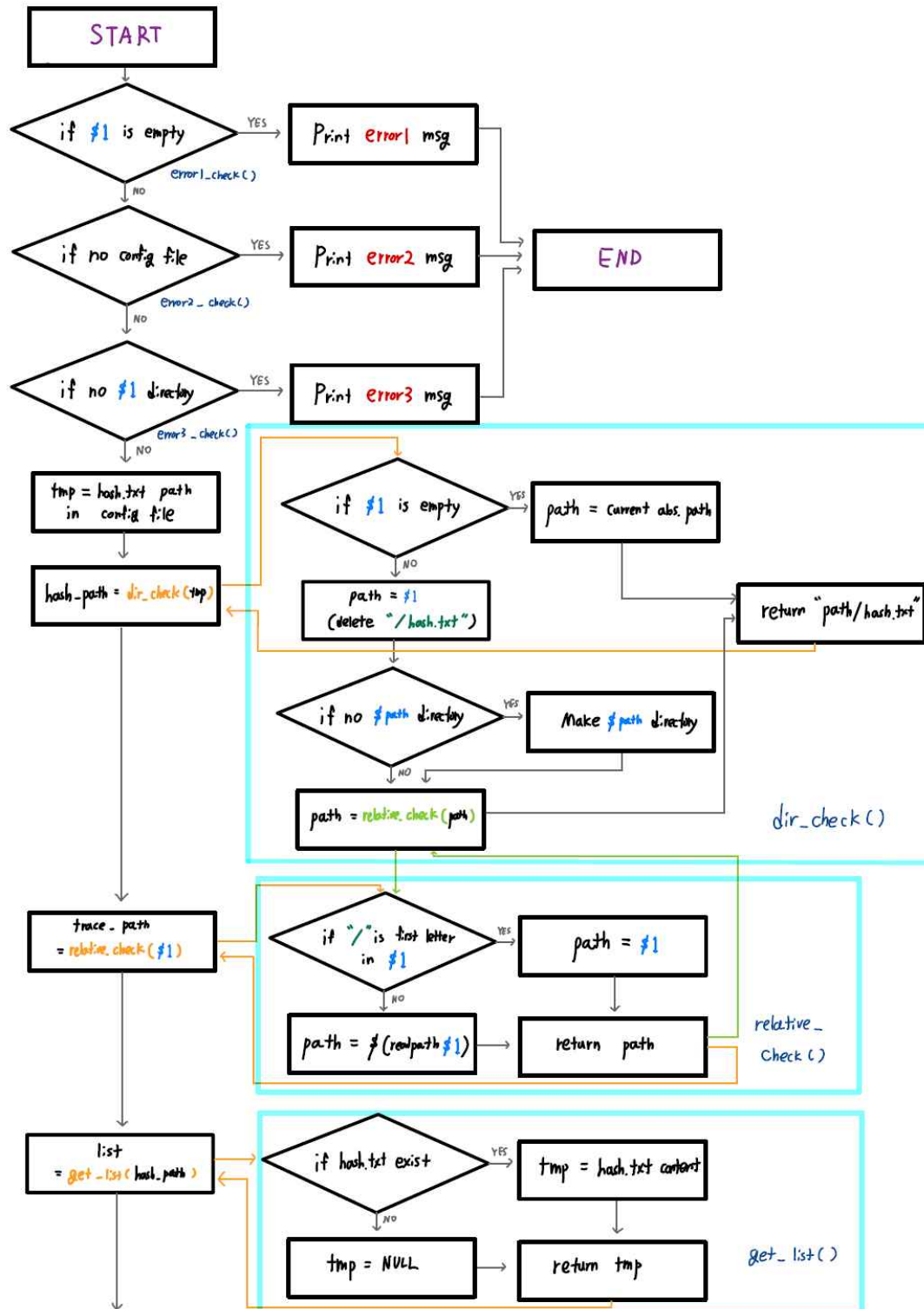
프로그램을 알아보기 전, 이 프로그램이 왜 필요하고 목적이 무엇인지를 간단하게 생각해 보았는데, 이를 간략히 살펴보고자 한다. 무결성이란 파일이나 데이터가 손상되지 않고 원래 상태를 정상적으로 유지함을 의미하는데, 이를 항상 보증해야만 정상적인 시스템 운영이 가능하므로 데이터의 무결성 검사는 필수적이다. 무결성 검사로 인해 얻는 다른 이점들은 아래와 같다.

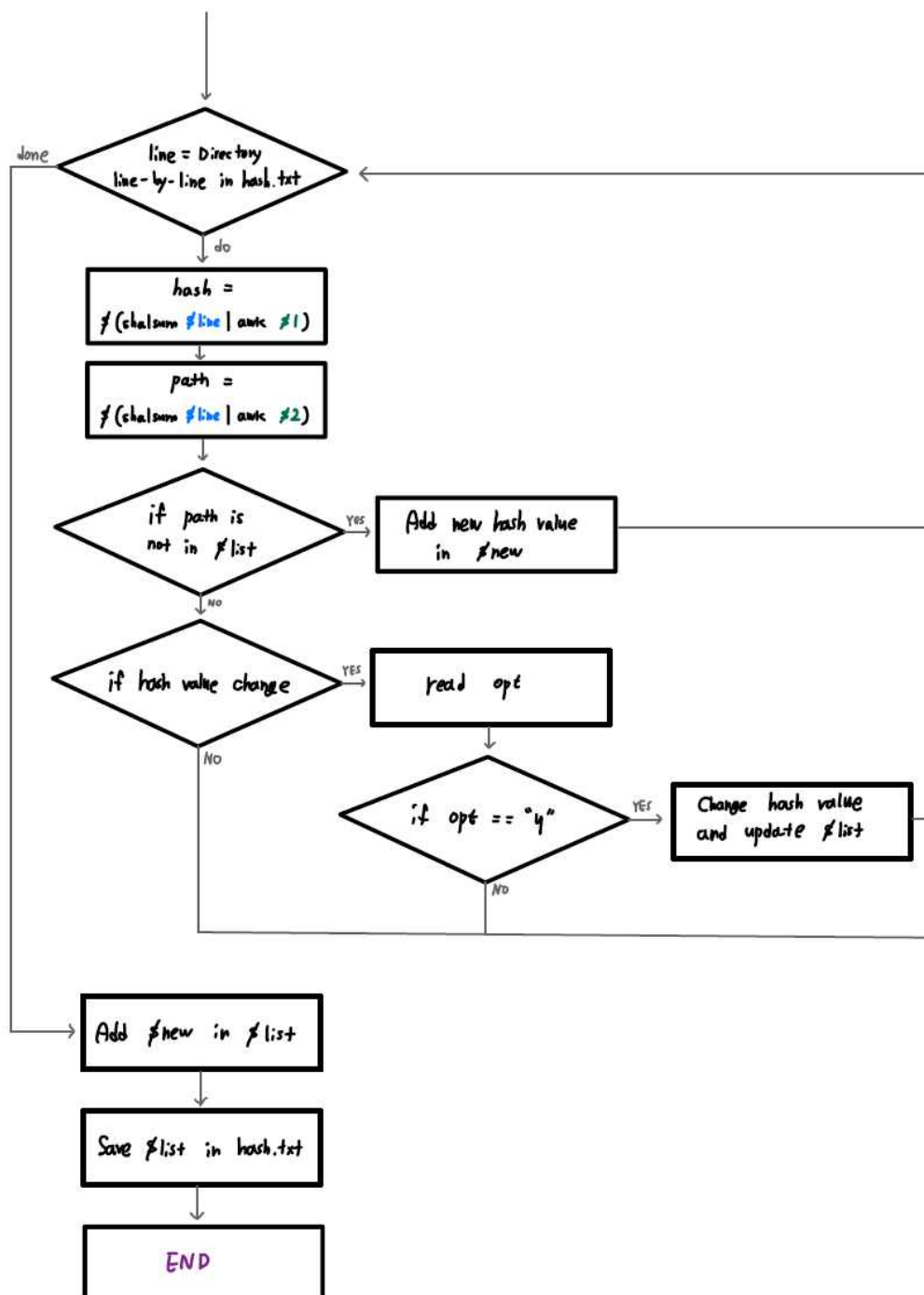
1. **신뢰성** : 데이터가 무결함을 증명함으로써 시스템 사용자에게 신뢰를 준다.
2. **데이터 복구** : 무결성 검사는 데이터 손실을 빠르게 감지하고 정상 상태로 복구하도록 돕는다. 그래서 데이터 손실을 최소화하고 시스템 가용성을 높일 수 있다.
3. **보안 강화** : 간혹 악의적인 사용자가 파일을 손상시키거나 변조할 수 있는데, 이러한 공격들을 방지하고 방어할 수 있는 수단으로 무결성 검사가 대표적이다. 이를 통해 문제에 대한 빠른 피드백을 하여 보안을 강화하는 데에도 도움이 된다.

이러한 무결성 검사에는 대표적으로 해시 함수를 사용한다. 파일의 해시 값들을 계산하고 저장하는데, 나중에 파일이 변경되었는지 확인할 때 다시 파일의 해시값을 계산하고 저장된 해시값과 비교하는 것이다. 바로 이것이 이번 프로그램의 핵심인데 해시값 계산을 위한 해시 함수로 쉘 스크립트의 sha1sum를 사용한다. sha1sum은 SHA-1 해시함수를 사용하여 해시 값을 계산하는 명령어이다. 요즘은 SHA-1보다 더 큰 비트 길이를 이용하여 해시값을 계산하는 SHA-256이나 SHA-3등이 있지만 과제의 안내대로 우선 SHA-1을 사용해보도록 한다.

3. Design

현재 파트에서는 프로그램의 전체적인 구조를 flow-chart로 간략히 소개하고, 구조에 대한 자세한 설명은 다음 파트인 Implementation에서 살펴보도록 한다.





4. Implementation

이전 파트에서 볼 수 있듯, 프로그램 코드는 main 로직과 6개의 추가적인 함수로 구성되어있다. 우선 3가지의 error_check 함수는 예외처리를 위한 함수이다. error1은 매개변수(\$1)가 존재하지 않을 때, error2는 config파일을 인식할 수 없을 때, error3은 매개변수로 받은 경로가 존재하지 않을 때이다. 이 3가지의 예외가 발생하면 무결성 검사 자체를 진행할 수 없으므로 예외가 발생한 순간 프로그램을 종료해야 한다. 나머지 함수는 dir_check, relative_check, get_list인데, 이들은 무결성 검사에 들어가기 전 필요한 작업을 진행한다. dir_check는 config파일에 적힌 hash.txt 경로가 실제로 존재하는지를 확인하고, 존재하지 않으면 경로를 만들어준다. relative_check는 매개변수 \$1나 config파일 내 hash.txt의 경로가 상대경로일 경우 절대경로로 바꿔주는 기능을 한다. get_list는 hash.txt가 이미 존재할 시 이 내용들을 변수에 저장하여 return해준다. 지금까지 설명한 6가지 기능들은 프로그램의 핵심 로직인 무결성 검사를 단순히 보조해 주는 역할이므로 main에 같이 작성하지 않고 함수로 따로 만들어 주었다. 이 과정을 모두 거친다면 변수 trace_path에는 최종 hash.txt의 절대경로가, 변수 list에는 기존 해시값 정보가 저장된다. 이제 trace_path 경로의 모든 하위경로에 존재하는 파일들에 대해 무결성 검사를 진행한다. 앞서 말한 것처럼 sha1sum을 이용해 해시값을 계산하며 정보가 없으면 추가, 정보가 있는데 해시값 변화가 생기면 사용자와 상호작용하여 해시값을 검토한다. 무결성 검사가 종료되면 업데이트한 내용을 hash.txt에 저장하여 끝낸다. 구현의 특이점으로, 코드 70번째 줄에 <3과 101쪽 done 옆에 3<을 볼 수 있다. 원래는 while문에서 이미 hash.txt를 read로 읽고 있어 기본 파일 디스크립터를 이미 사용하는 상태라 while문 내에서 사용자에게 입력받을 수 없다. 이 프로그램에서는 해시값이 변경되었을 때 사용자가 해시값을 바꿀건지 안바꿀건지를 선택해야 하므로 사용자에게 y or n을 입력받아야 한다. 따라서 hash.txt를 읽는 파일 디스크립터를 3번으로 바꿔주면 충돌을 막을 수 있다.

5. Test and Results

기존 환경에 존재하는 /tmp/hash.txt는 접근 권한 문제로 인해 확인할 수가 없어 config 파일에는 다른 경로를 적어주었다. 테스트 케이스는 아래와 같다.

매개변수 \$1	config에 적힌 hash 경로
공백	-

↳ Test1) error1 테스트

매개변수 \$1	config에 적힌 hash 경로
/home/ubuntu	config 파일 존재X

↳ Test2) error2 테스트

매개변수 \$1	config에 적힌 hash 경로
/home/ubuntu/noDirectory	-

↳ Test3) 테스트

매개변수 \$1	config에 적힌 hash 경로
/home/ubuntu/hw1/test	공백

↳ Test4) config에 아무것도 적혀있지 않을 경우

매개변수 \$1	config에 적힌 hash 경로
test/test1/tmp2/hello	noDirectory/hash.txt

↳ Test5) 매개변수가 파일명(상대경로) + config에 적힌 상대경로가 존재X

매개변수 \$1	config에 적힌 hash 경로
/home/ubuntu/hw1/test/yys.txt	/home/ubuntu/hw1/noDirectory

↳ Test6) 매개변수가 파일명(절대경로) + config에 적힌 절대경로가 존재X + config에 적힌 경로에 hash.txt가 안적혀있는 경우

매개변수 \$1	config에 적힌 hash 경로
test/test1	tmp/hash.txt

↳ Test7) 매개변수와 hash경로 모두 정상적인 상대경로 + 해시값 변경 테스트

매개변수 \$1	config에 적힌 hash 경로
/home/ubuntu	/home/ubuntu/hw1/tmp/hash.txt

↳ Test8) 매개변수와 hash경로 모두 정상적인 절대경로 (일반적인 경우)

● Test1 Result

```
ubuntu@yys:~$ hw1/check_integrity.sh
Usage: hw1/check_integrity.sh [directory]
ubuntu@yys:~$ cd hw1
ubuntu@yys:~/hw1$ ./check_integrity.sh
Usage: ./check_integrity.sh [directory]
```

매개변수가 없으므로 error1 메시지 출력 후 프로그램 종료

● Test2 Result

```
ubuntu@yys:~/hw1$ ls
check_integrity.sh test tmp
ubuntu@yys:~/hw1$ ./check_integrity.sh /home/ubuntu
Config does not exist.
```

● Test3 Result

```
ubuntu@yys:~$ ls
cpt5 cpt5.tar cpt6 cpt7 cpt8 execfile.sh hw1 userlist.sh
ubuntu@yys:~$ cd hw1
ubuntu@yys:~/hw1$ ls
check_integrity.sh config test tmp
ubuntu@yys:~/hw1$ ./check_integrity.sh /home/ubuntu/noDirectory
/home/ubuntu/noDirectory does not exist.
ubuntu@yys:~/hw1$
```

매개변수로 받은 디렉터리가 존재하지 않으므로 error3 메시지 출력 후 프로그램 종료

● Test4 Result

```
hw1 > ≡ config
1
```

비어있는 config 파일 (/home/ubuntu/hw1/config)

```
ubuntu@yys:~/hw1$ ./check_integrity.sh /home/ubuntu/hw1/test

[NEW] /home/ubuntu/hw1/test/yys.txt (568f581ef65bec99607d0d001af70c3dead98e15)
[NEW] /home/ubuntu/hw1/test/test1/bce.sh (feb56457f9659937a2afa2f22faf344515035130)
[NEW] /home/ubuntu/hw1/test/test1/script6-1.sh (27033db29a85ae2542a939c3d257a56cf070962e)
[NEW] /home/ubuntu/hw1/test/test1/tmp2/userlist.sh (4ad981f74cacf74880ef67571293610e0e93581f)
[NEW] /home/ubuntu/hw1/test/test1/tmp2/hello (6adfb183a4a2c94a2f92dab5ade762a47889a5a1)
[NEW] /home/ubuntu/hw1/test/test1/gugudan.sh (598eaff66a67904d6f07fa5f62a96503513dabb7)
[NEW] /home/ubuntu/hw1/test/test1/seats.txt (141df7e0f24cb94efda4a02f1e0c2212f729e302)
[NEW] /home/ubuntu/hw1/test/test1/script6-2.sh (91628345061c5e3aaf2073be1f362e36d6870585)
[NEW] /home/ubuntu/hw1/test/test1/forin1.sh (1c03b8299e11e2e7e33539e72cec9696735c3a6f)

Integrity check completed.
```

/home/ubuntu/hw1/test 내 파일들의 무결성 검사

```
hw1 > ≡ hash.txt
1 568f581ef65bec99607d0d001af70c3dead98e15 /home/ubuntu/hw1/test/yys.txt
2 feb56457f9659937a2afa2f22faf344515035130 /home/ubuntu/hw1/test/test1/bce.sh
3 27033db29a85ae2542a939c3d257a56cf070962e /home/ubuntu/hw1/test/test1/script6-1.sh
4 4ad981f74cacf74880ef67571293610e0e93581f /home/ubuntu/hw1/test/test1/tmp2/userlist.sh
5 6adfb183a4a2c94a2f92dab5ade762a47889a5a1 /home/ubuntu/hw1/test/test1/tmp2/hello
6 598eaff66a67904d6f07fa5f62a96503513dabb7 /home/ubuntu/hw1/test/test1/gugudan.sh
7 141df7e0f24cb94efda4a02f1e0c2212f729e302 /home/ubuntu/hw1/test/test1/seats.txt
8 91628345061c5e3aaf2073be1f362e36d6870585 /home/ubuntu/hw1/test/test1/script6-2.sh
9 1c03b8299e11e2e7e33539e72cec9696735c3a6f /home/ubuntu/hw1/test/test1/forin1.sh
10
```

정상적으로 해시값들이 /home/ubuntu/hw1/hash.txt에 저장된 것을 확인

● Test5 Result

```
hw1 > ≡ config
1 noDirectory/hash.txt
2
```

config에 존재하지 않는 상대경로가 저장되어있는 상태

```
● ubuntu@yys:~/hw1$ ls
check_integrity.sh config test tmp
● ubuntu@yys:~/hw1$ ./check_integrity.sh test/test1/tmp2/hello

[NEW] /home/ubuntu/hw1/test/test1/tmp2/hello (6adfb183a4a2c94a2f92dab5ade762a47889a5a1)

Integrity check completed.

● ubuntu@yys:~/hw1$ ls
check_integrity.sh config noDirectory test tmp
```

존재하지 않는 디렉터리가 만들어지고 해시값이 저장되는 과정

```
hw1 > noDirectory > ≡ hash.txt
1 6adfb183a4a2c94a2f92dab5ade762a47889a5a1 /home/ubuntu/hw1/test/test1/tmp2/hello
2
```

해시값이 정상적으로 noDirectory/hash.txt에 저장된 것을 확인

● Test6 Result

```
hw1 > ≡ config
1  /home/ubuntu/hw1/noDirectory
2  
```

config에 존재하지 않는 절대경로가 저장되어 있고 /hash.txt가 작성 안된 상태

```
● ubuntu@yys:~/hw1$ ls
check_integrity.sh config test tmp
● ubuntu@yys:~/hw1$ ./check_integrity.sh /home/ubuntu/hw1/test/yys.txt

[NEW] /home/ubuntu/hw1/test/yys.txt (568f581ef65bec99607d0d001af70c3dead98e15)

Integrity check completed.

● ubuntu@yys:~/hw1$ ls
check_integrity.sh config noDirectory test tmp
```

존재하지 않는 디렉터리가 만들어지고 해시값이 저장되는 과정

```
hw1 > noDirectory > ≡ hash.txt
1  568f581ef65bec99607d0d001af70c3dead98e15 /home/ubuntu/hw1/test/yys.txt
2  
```

해시값이 정상적으로 noDirectory/hash.txt에 저장된 것을 확인

● Test7 Result

```
hw1 > ≡ config
1 tmp/hash.txt
2
```

config 파일에 적힌 상대경로

```
hw1 > test > test1 > tmp2 > ≡ hello
1 helloworld

hw1 > test > test1 > ≡ seats.txt
1 0 1 0 0 0 0 0
2 0 0 0 0 1 0 0 0
3 0 0 0 0 0 0 0 0
4
```

기존 hello와 seats.txt의 파일내용

```
ubuntu@yys:~/hw1$ ./check_integrity.sh test/test1

[NEW] /home/ubuntu/hw1/test/test1/bce.sh (feb56457f9659937a2afa2f22faf344515035130)
[NEW] /home/ubuntu/hw1/test/test1/script6-1.sh (27033db29a85ae2542a939c3d257a56cf070962e)
[NEW] /home/ubuntu/hw1/test/test1/tmp2/userlist.sh (4ad981f74cacf74880ef67571293610e0e93581f)
[NEW] /home/ubuntu/hw1/test/test1/tmp2/hello (6adfb183a4a2c94a2f92dab5ade762a47889a5a1)
[NEW] /home/ubuntu/hw1/test/test1/gugudan.sh (598eaff66a67904d6f07fa5f62a96503513dabb7)
[NEW] /home/ubuntu/hw1/test/test1/seats.txt (141df7e0f24cb94efda4a02f1e0c2212f729e302)
[NEW] /home/ubuntu/hw1/test/test1/script6-2.sh (91628345061c5e3aaf2073be1f362e36d6870585)
[NEW] /home/ubuntu/hw1/test/test1/forin1.sh (1c03b8299e11e2e7e33539e72cec9696735c3a6f)

Integrity check completed.

hw1 > tmp > ≡ hash.txt
1 feb56457f9659937a2afa2f22faf344515035130 /home/ubuntu/hw1/test/test1/bce.sh
2 27033db29a85ae2542a939c3d257a56cf070962e /home/ubuntu/hw1/test/test1/script6-1.sh
3 4ad981f74cacf74880ef67571293610e0e93581f /home/ubuntu/hw1/test/test1/tmp2/userlist.sh
4 6adfb183a4a2c94a2f92dab5ade762a47889a5a1 /home/ubuntu/hw1/test/test1/tmp2/hello
5 598eaff66a67904d6f07fa5f62a96503513dabb7 /home/ubuntu/hw1/test/test1/gugudan.sh
6 141df7e0f24cb94efda4a02f1e0c2212f729e302 /home/ubuntu/hw1/test/test1/seats.txt
7 91628345061c5e3aaf2073be1f362e36d6870585 /home/ubuntu/hw1/test/test1/script6-2.sh
8 1c03b8299e11e2e7e33539e72cec9696735c3a6f /home/ubuntu/hw1/test/test1/forin1.sh
9
```

파일의 해시값이 저장된 모습, 이제 두 파일의 내용을 바꾼 뒤 테스트

(hello : 4번째줄, seats.txt : 6번째줄)


```

hw1 > test > test1 > tmp2 > ≡ hello
1  hello, world!!!

hw1 > test > test1 > ≡ seats.txt
1  0 1 0 0 0 0 0 0
2  0 0 0 0 1 0 0 0
3  0 0 0 0 0 0 0 0
4  1 1 1 1 1 1 1 1
5

```

변경된 hello와 seats.txt의 내용

```

ubuntu@yys:~/hw1$ ./check_integrity.sh test/test1

[WARN] /home/ubuntu/hw1/test/test1/tmp2/hello
(old: 6adfb183a4a2c94a2f92dab5ade762a47889a5a1, new: 7deffbce6a3601b14a18e90f88550f88ae52165c)

=====

/home/ubuntu/hw1/test/test1/tmp2/hello is changed.

Do you want to update the hash of /home/ubuntu/hw1/test/test1/tmp2/hello (y|n)?
y

y : Change hash value

=====

[WARN] /home/ubuntu/hw1/test/test1/seats.txt
(old: 141df7e0f24cb94efda4a02f1e0c2212f729e302, new: e5a5afc531f1b51cae531e00028204acef302745)

=====

/home/ubuntu/hw1/test/test1/seats.txt is changed.

Do you want to update the hash of /home/ubuntu/hw1/test/test1/seats.txt (y|n)?
n

n : No change

=====

Integrity check completed.

hw1 > tmp > ≡ hash.txt
1  feb56457f9659937a2afa2f22faf344515035130 /home/ubuntu/hw1/test/test1/bce.sh
2  27033db29a85ae2542a939c3d257a56cf070962e /home/ubuntu/hw1/test/test1/script6-1.sh
3  4ad981f74cacf74880ef67571293610e0e93581f /home/ubuntu/hw1/test/test1/tmp2/userlist.sh
4  7deffbce6a3601b14a18e90f88550f88ae52165c /home/ubuntu/hw1/test/test1/tmp2/hello
5  598eaff66a67904d6f07fa5f62a96503513dabb7 /home/ubuntu/hw1/test/test1/gugudan.sh
6  141df7e0f24cb94efda4a02f1e0c2212f729e302 /home/ubuntu/hw1/test/test1/seats.txt
7  91628345061c5e3aaf2073be1f362e36d6870585 /home/ubuntu/hw1/test/test1/script6-2.sh
8  1c03b8299e11e2e7e33539e72cec9696735c3a6f /home/ubuntu/hw1/test/test1/forin1.sh
9

```

hello는 해시값 변경을 허용하고, seats.txt만 허용한 모습

정상적으로 hello의 해시값만 변경된 모습을 확인 (hello : 4번째줄, seats.txt : 6번째줄)

● Test8 Result

```
hw1 > ≡ config
1 /home/ubuntu/hw1/tmp/hash.txt
2
```

config의 내용

```
[NEW] /home/ubuntu/.bash_history (6d2730c759c95defbba92401f943e53adfe14e28)
[NEW] /home/ubuntu/cpt8/act1.sh (da39a3ee5e6b4b0d3255bfef95601890afd80709)
[NEW] /home/ubuntu/cpt8/rpc/server.py (88eb1a918e9177f4c1a9d477f422d7243d606722)
[NEW] /home/ubuntu/cpt8/rpc/client.py (da39a3ee5e6b4b0d3255bfef95601890afd80709)
[NEW] /home/ubuntu/cpt8/act2.sh (da39a3ee5e6b4b0d3255bfef95601890afd80709)
[NEW] /home/ubuntu/cpt8/flask/server.py (c9169db9daa6534633ec1080c66b35e2b5453f7d)
[NEW] /home/ubuntu/cpt8/flask/client.py (c2d70f5a27dc0b2f253afb04f0199ea56a8269e4)
[NEW] /home/ubuntu/.ssh/authorized_keys (2db627780c86f7e230f09f44cf614cd50ac6c002)
[NEW] /home/ubuntu/.profile (2b9ee6d446f8f9ffccaab42b6df5649f749a9a07)
```

Integrity check completed.

```
hw1 > tmp > ≡ hash.txt
1 feb56457f9659937a2afa2f22faf344515035130 /home/ubuntu/cpt5/bce.sh
2 666267416b96d1bb6fadeaaa1105a4e941960eb2 /home/ubuntu/cpt5/while3.sh
3 7386eb6b20a7430b44c1cfa1524261947c766f57 /home/ubuntu/cpt5/while2.sh
4 141df7e0f24cb94efda4a02f1e0c2212f729e302 /home/ubuntu/cpt5/seats.txt
5 0dfd4e09fa4a3726fe4421725f92d8278ef5e450 /home/ubuntu/cpt5/forin2.sh
6 8d69b2c69bc2c71575861a181fa9804b0fd36a38 /home/ubuntu/cpt5/script1.sh
7 3ad7e0e0c7ed527b0b393c06876e2cb834b66642 /home/ubuntu/cpt5/func2.sh
8 67875b9f869d6afae4f1fa4ead001b0457047d9f /home/ubuntu/cpt5/script5-2.sh
9 554cd4ad8bbeba6436d615647a4579b6ccc28b13 /home/ubuntu/cpt5/func1.sh
10 bfbaf2be4d8da3f6e9f63d87bf663972a5bf4fd5 /home/ubuntu/cpt5/script5-1.sh
```

●
●
●

```
14925 4a300c4314c7f4003ce87eb38efb6f6a9c459b27 /home/ubuntu/.local/lib/python3.10/
14926 6d2730c759c95defbba92401f943e53adfe14e28 /home/ubuntu/.bash_history
14927 da39a3ee5e6b4b0d3255bfef95601890afd80709 /home/ubuntu/cpt8/act1.sh
14928 88eb1a918e9177f4c1a9d477f422d7243d606722 /home/ubuntu/cpt8/rpc/server.py
14929 da39a3ee5e6b4b0d3255bfef95601890afd80709 /home/ubuntu/cpt8/rpc/client.py
14930 da39a3ee5e6b4b0d3255bfef95601890afd80709 /home/ubuntu/cpt8/act2.sh
14931 c9169db9daa6534633ec1080c66b35e2b5453f7d /home/ubuntu/cpt8/flask/server.py
14932 c2d70f5a27dc0b2f253afb04f0199ea56a8269e4 /home/ubuntu/cpt8/flask/client.py
14933 2db627780c86f7e230f09f44cf614cd50ac6c002 /home/ubuntu/.ssh/authorized_keys
14934 2b9ee6d446f8f9ffccaab42b6df5649f749a9a07 /home/ubuntu/.profile
14935
```

/home/ubuntu 내 모든 파일들의 해시값이 정상적으로 hash.txt에 저장된 모습

6. Conclusion

무결성 검사 프로그램을 구현해보며 무결성 검사가 파일들의 해시값을 계산하고 이들을 비교하며 진행하는 검사라는 것을 알았다. 이를 통해 파일의 데이터 손상이나 위변조가 일어났을 때 손쉽게 감지가 가능하다는 것을 배울 수 있었다. 이번 과제는 코드를 짜기 전 해시에 대해 알아보고 왜 무결성 검사가 중요한지에 대해 간단히 프리뷰했는데, 이것이 과제를 하는 데 확실한 동기부여가 됐다. 그리고 단순히 시스템 밖에서 코딩하는 것에 그쳤던 과거와 다르게, 파일들의 경로들을 직접 다루면서 시스템 내의 파일에 접근하는 프로그램을 처음 만들어 봤다. 약간 어렵다는 생각도 했지만 얻어간 것이 훨씬 많은 새로운 경험이었다.

또한 과제를 시작하면서 잡았던 목표가 다양한 리눅스 명령어들이 손에 익도록 연습하고 최종적으로 짜임새 있고 간결한 쉘 스크립트를 작성하는 것이었다. 중간고사 이전까지의 내 자신과 비교하면 확실히 발전했다는 느낌을 받았지만, 아직도 불필요한 구조나 코드가 있어 부족하다고 생각한다. 대표적으로 sha1sum을 2번 진행한 것과 변수 old_hash와 flag를 따로 나눈 코드인데, 분명 더 효율적인 방법이 있을거라 믿어 개선하려다 아쉽게 실패하였다. 다음 과제에서는 이러한 내용들을 피드백하여 더욱 정교한 프로그램을 만들고 싶다.