설계과제 2 명세 : Soongsil sdup

Linux System Programming, School of CSE, Soongsil University, Spring 2022

○ 개요

- ssu_sdup은 시스템 내 존재하는 동일(중복)한 파일을 찾고 삭제하는 프로그램

○ 목표

- 유닉스/리눅스 시스템에서 제공하는 시스템 자료구조와 시스템콜 및 라이브러리 함수를 이용하여 프로그램을 작성함으로써 시스템 프로그래밍 설계 및 응용 능력 향상

○ 팀 구성

- 개인별 프로젝트

○ 보고서 제출 방법

- 설계과제는 "보고서.hwp"(개요, 상세설계 구현방법, 결과 및 소스코드와 실행결과가 함께 있는 워드 (hwp 또는 MS-Word) 파일))와 "소스코드"(makefile, obj, *.c, *.h 등 컴파일하고 실행하기 위한 모든 파일)를 포함시켜야 함
- 과제 2번은 makefile을 필수로 제출해야하며, ssu_sdup.c ssu_find-md5.c, ssu_find-sha1.c, ssu_help.c 는 기본적으로 포함해야 함. 기타 학생이 정의한 헤더 파일 등을 포함시켜도 됨. 아래 각 C 프로그램은 학생들이 추가로 다른 내용을 포함시켜도 됨
  - ✓ ssu_sdup.c : 프롬프트 출력 및 내장명령어 호출을 위한 C 프로그램
  - ✓ ssu_find-md5.c : ssu_sdup의 내장명령어 중 md5()를 사용하여 fmd5 기능을 수행하기 위한 C 프로그램 (동일한 파일 탐색 및 삭제)
  - ✓ ssu_find-sha1.c : ssu_sdup의 내장명령어 중 sha1()을 사용하여 fsha1 기능을 수행하기 위한 C 프로그램 (동일한 파일 탐색 및 삭제)
  - ✓ ssu_help.c : ssu_sdup의 내장명령어 중 help 기능을 수행하기 위한 C 프로그램 (내장명령어 사용법 출력)
- "#P2_학번_버전.zip"(예. #P2_20160000_V1.zip)형태로 (zip프로그램으로 압축하여) 파일 이름을 명명하고, classroom.google.com에 마감일까지 제출. 단, 서버 다운으로 이메일(2022.oslab1@gmail.com과 2022.oslab2@gmail.com)로 제출 시 이메일 제목은 "#P설계과제번호_학번_버전"이어야 하고, 내용은 없이 "#P설계과제번호_학번_버전.zip)만 첨부하면 됨. 버전 이름은 대문자 V와 함께 integer를 1부터 incremental 증가시키면서 부여하면 됨. (예. V1, V2, V3 ...)
- 압축파일 내 "보고서" 디렉토리와 "소스코드" 디렉토리 2개 만들어야 함. 소스코드 내 makefile이 포함되어야 함.
- 제출한 압축 파일을 풀었을 때 해당 디렉토리에서 컴파일 및 실행이 되어야 함(특정한 디렉토리에서 실행해야 할 경우는 제외). 해당 디렉토리에서 컴파일이나 실행되지 않을 경우, 설계과제 제출 방법(파일명, 디렉토리명, 컴파일 시에 포함되어야 할 파일 등)을 따르지 않는 경우 해당 과제 배점의 50% 감점
- 기타 내용은 강의계획서 참고

○ 제출 기한

- 4월 20일(수) 오후 11시 59분 59초 (서버 시간 기준 1시간까지 지연 허용), 최대 3일까지 지연 제출 가능, 단 1일 지연 제출마다 30% 감점. 4일째는 0점 처리

○ 보고서 양식

- 보고서는 다음과 같은 양식으로 작성

```
1. 과제 개요 (1점) // 위 개요를 더 상세하게 작성
2. 구현 플랫폼 (2점) // Linux,, MacOS, 커널 버전(uname -a) 등
3. 상세 설계 (5점) // 함수 기능별 흐름도(순서도) 반드시 포함
4. 구현 방법 설명 (5점) // 함수 프로토타입 반드시 포함
5. 실행 결과 (2점) // 테스트 프로그램의 실행 결과 캡쳐 및 분석
6. 소스코드 (85점) // 주석, 컴파일 여부 (10점), 실행 여부 (75점)
```

○ ssu_sdup 프로그램 기본 사항

- ssu_sdup 은 지정한 디렉토리 내에서 사용자가 입력한 조건(파일 확장자, 파일 크기 범위)에 맞춰 동일한(중복된) 정규 파일을 찾고 삭제하는 프로그램으로 동일한 정규 파일은 파일의 이름이 달라도 파일의 크기와 내용이 동일한 경우를 말함. ssu_sdup은 텍스트 파일 뿐만 아니라 모든 형태의 바이너리 파일을 대상으로 함
- ssu_sdup에 내장된 fmd5, fsha1 명령어를 통해 네 번째 인자의 시작 디렉토리 아래 모든 서브디렉토리까지 재귀적으로 탐색하여, 첫 번째 인자로 주어진 파일과 동일한(중복된) 파일을 찾는데, 두 번째 인자와 세 번째 인자 크기 사이의 파일만을 대상으로 찾음
- 동일한(중복) 파일이 존재하는 경우 사용자가 입력한 옵션에 따라 동일한(중복) 파일 삭제
- ssu_sdup은 내장명령어(ssu_find-md5.c의 fmd5, ssu_find-sha1.c의 fsha1, ssu_help.c의 help)를 위한 프로세스를 생성(fork())하고 이를 실행(exec() 류 함수)시켜 줌 (exit 명령어 제외)
- ssu_sdup에 내장된 fmd5, fsha1 명령어는 링크드 리스트로 동일한(중복) 파일 리스트를 관리해야 함
  - ✓ 리눅스 상에서 파일 경로의 최대 크기는 4,096 바이트이며, 파일 이름의 최대 크기는 255 바이트임
- system() 사용 불가

○ 설계 및 구현

- 가) ssu_sdup
  - ✓ 프롬프트 출력
  - ✓ 프롬프트 모양 : "학번〉 " 문자 출력 (예. 20201234〉)
  - ✓ 프롬프트에서는 ssu_sdup의 내장 명령어인 fmd5, fsha1, exit, help 명령어만 실행
  - ✓ 이외 명령어 실행 시 자동으로 help를 실행시킨 것과 동일한 결과가 출력되어야 함
  - ✓ 엔터 키만 입력 시 프롬프트 재출력
- 나) 내장명령어 1, fmd5
  - ✓ 사용법 : fmd5 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY]
    - 지정한 디렉토리(TARGET_DIRECTORY) 아래 모든 디렉토리에서 지정한 파일 확장자(FILE_EXTENSION)을 갖는 특정 크기 사이(MINSIZE 이상 MAXSIZE 이하) 동일한(중복) 파일을 찾아 리스트로 출력
      - 지정한 디렉토리의 하위 디렉토리를 재귀적으로 탐색하여 동일한(중복) 파일을 찾음.
      - 단, 하위 디렉토리 탐색은 BFS ~~DFS~~ 알고리즘을 사용
    - "동일한(중복) 파일"은 파일의 해시값이 같은 정규 파일(텍스트, 바이너리 모두 포함)임
      - 파일별 해시값 비교 예시
      - 예시 1~4는 각각 ~/test/a, ~/test/b, ~/test/c, ~/test/d 파일의 내용을 보여줌
      - 예시 5는 ~/test/a, ~/test/b, ~/test/c, ~/test/d 파일의 크기를 보여줌
      - 예시 6은 ~/test/a, ~/test/b, ~/test/c, ~/test/d 파일의 해시값을 보여줌(과제 구현을 위해 설명하는 예제이므로 본 과제에서는 구현하지 않음). a 파일과 d 파일의 해시값이 같다는 것을 알 수 있으며, 이는 동일한(중복) 파일임을 나타냄

(예시 1) a 파일 내용

```
1 Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operati
  ng system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packa
  ged in a Linux distribution.
2 Distributions include the Linux kernel and supporting system software and libraries, many of which
   are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but
   the Free Software Foundation uses the name "GNU/Linux" to emphasize the importance of GNU softwar
  e, causing some controversy.
```

(예시 2) b 파일 내용 - a 파일과 비교하면 한 문자가 다른 파일

```
1 Pinux is a family of open-source Unix-like operating systems based on the Linux kernel, an operati
  ng system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packa
  ged in a Linux distribution.
2 Distributions include the Linux kernel and supporting system software and libraries, many of which
   are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but
   the Free Software Foundation uses the name "GNU/Linux" to emphasize the importance of GNU softwar
  e, causing some controversy.
```

(예시 3) c 파일 내용 – a 파일과 비교하면 일부 내용이 추가된 파일

```
1 Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operati
  ng system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packa
  ged in a Linux distribution.
2 Distributions include the Linux kernel and supporting system software and libraries, many of which
   are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but
   the Free Software Foundation uses the name "GNU/Linux" to emphasize the importance of GNU softwar
  e, causing some controversy.
3 Popular Linux distributions include Debian, Fedora Linux, and Ubuntu. Commercial distributions inc
  lude Red Hat Enterprise Linux and SUSE Linux Enterprise. Desktop Linux distributions include a win
  dowing system such as X11 or Wayland, and a desktop environment such as GNOME or KDE Plasma. Distr
  ibutions intended for servers may omit graphics altogether, or include a solution stack such as LA
  MP. Because Linux is freely redistributable, anyone may create a distribution for any purpose.
```

(예시 4) d 파일 내용 – a 파일과 내용이 동일한 파일

```
1 Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operati
  ng system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packa
  ged in a Linux distribution.
2 Distributions include the Linux kernel and supporting system software and libraries, many of which
   are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but
   the Free Software Foundation uses the name "GNU/Linux" to emphasize the importance of GNU softwar
  e, causing some controversy.
```

(예시 5) ~/test/a, b, c, d 파일의 크기 비교

```
su@su-virtual-machine:~/test-hash$ ls -alH
total 52
drwxrwxr-x  2 su su  4096  3월 22 16:15 .
drwxr-xr-x 28 su su  4096  3월 22 16:15 ..
-rw-rw-r--  1 su su   547  3월 22 15:40 a
-rw-rw-r--  1 su su   547  3월 22 16:15 b
-rw-rw-r--  1 su su  1034  3월 22 15:33 c
-rw-rw-r--  1 su su   547  3월 22 16:15 d
```

(예시 6) ~/test/a, b, c, d 파일의 해시값 비교

```
su@su-virtual-machine:~/test-hash$ ./ssu_md5 a
05f8d01138ae7136df8c4ddd3f01e378
su@su-virtual-machine:~/test-hash$ ./ssu_md5 b
7c18e0cd5626833790abdb2cbb2fd1a4
su@su-virtual-machine:~/test-hash$ ./ssu_md5 c
aa50894bd6837536ad407d6297d1263a
su@su-virtual-machine:~/test-hash$ ./ssu_md5 d
05f8d01138ae7136df8c4ddd3f01e378
```

- 따라서, 동일한(중복) 파일 탐색 시, 크기가 동일한 파일을 찾은 후 같은 크기를 가진 파일 중 각 파일의 해시값을 비교하여 동일한(중복) 파일 리스트를 만듦
- 파일 이름이 다른 여러 개의 동일한(중복) 파일이 있을 수 있으며, 하나의 동일한(중복) 파일 리스트가 하나의 동일한(중복) 파일 세트가 됨. 예시 7에서 총 6개의 동일한(중복) 파일 리스트 세트를 확인할 수 있음 (Identical files #1 ~ #6, 총 6 개 세트)
- 여러 동일한(중복) 파일 세트가 존재할 때, 하나의 세트에는 여러 개의 동일한(중복) 파일들이 있을 수 있으며, 동일한(중복) 파일 세트는 '파일 크기'를 기준으로 작은 파일부터 큰 파일로 정렬해서 출력. 파일의 크기가 같은 세트는 절대 경로를 기준으로 루트 디렉토리에서부터 경로가 짧은 세트부터 출력. 경로의 길이가 동일하면 임의로 세트를 정렬해서 출력하면 됨 (순서 상관 없음). 예시 7 참고
- MD5 해시값을 구하기 위해 MD5(openssl/md5.h)를 사용
  - ☞ Linux, Ubuntu :"sudo apt-get install libssl-dev"로 라이브러리 설치 필요
  - ☞ Linux, Fedora :"sudo dnf-get install libssl-devel"로 라이브러리 설치 필요
  - ☞ MacOS : homebrew (https://brew.sh/ 참고) 설치 -〉 % brew install openssl
- MD5 관련 함수를 사용하기 위해 컴파일 시 "-lcrypto" 옵션 필요
- md5() 사용법은 https://www.openssl.org/docs/man1.1.1/man3/MD5.html 및 https://github.com/Chronic-Dev/openssl/blob/master/crypto/md5/md5.c 참고

✓ 첫 번째 인자 [FILE_EXTENSION]
- 탐색할 파일의 확장자
- "*" 입력 시, 모든 정규 파일을 대상으로 중복 파일 탐색
- "*.(확장자)" 입력 시, (확장자)인 정규 파일에서만 중복 파일 탐색 (예. *.jpg : jpg 확장자를 가진 파일에서만 중복 파일 탐색)
- [FILE_EXTENSION] 인자에 입력이 없거나, "*", "*.(확장자)" 외 다른 입력이 들어오면 에러 처리 후 프롬프트 출력

✓ 두 번째 인자 [MINSIZE]
- 탐색할 파일의 최소 크기
- [MINSIZE] 인자에 바이트, KB, MB, GB 단위가 가능하며, 단위 미입력 시 기본 설정은 바이트 단위임 (단위는 소문자

도 가능해야 하며, 공백으로 구분하지 않음)
- [MINSIZE] 인자에 KB, MB, GB 단위는 실수도 가능해야 함 (실수의 소숫점 이하 자리는 KB는 3자리 이상, MB는 6자리 이상, GB는 9자리 이상이 표시되면 됨. 그 이상의 자리는 의미 없음)
- [MINSIZE] 인자에 '~' 입력할 경우, [MAXSIZE] 이하인 모든 파일에서 중복 파일 탐색
- [MINSIZE] 인자에 입력이 없거나, 숫자(실수 포함)나 '~' 외 다른 입력이 들어오면 에러 처리 후 프롬프트 출력

✓ 세 번째 인자 [MAXSIZE]
- 탐색할 파일의 최대 크기
- [MINSIZE] 인자에 바이트, KB, MB, GB 단위가 가능하며, 단위 미입력 시 기본 설정은 바이트 단위임 (단위는 소문자도 가능해야 하며, 공백으로 구분하지 않음)
- [MINSIZE] 인자에 KB, MB, GB 단위는 실수도 가능해야 함 (실수의 소숫점 이하 자리는 KB는 3자리 이상, MB는 6자리 이상, GB는 9자리 이상이 표시되면 됨. 그 이상의 자리는 의미 없음)
- [MAXSIZE] 인자에 '~'를 입력할 경우, [MINSIZE] 이상인 모든 파일에서 중복 파일 탐색
- [MAXSIZE] 인자에 입력이 없거나, 숫자(실수 포함)나 '~' 외 다른 입력이 들어오면 에러 처리 후 프롬프트 출력
- [MINSIZE]와 [MAXSIZE] 모두 '~'이면 크기 제한 없이 중복 파일 탐색

✓ 네 번째 인자 [TARGET_DIRECTORY]
- 탐색할 디렉토리 경로(절대경로와 "~(홈 디렉토리)"를 포함한 상대경로 모두 가능해야 함)
- 인자에 루트("/")를 입력할 경우, 파일 탐색 시 권한 문제가 발생할 수 있으므로 root 권한으로 실행해야 함
- 루트 디렉토리부터 탐색 시, "proc"과 "run", "sys" 디렉토리는 제외하여 탐색
- [TARGET_DIRECTORY] 인자에 입력이 없거나, 디렉토리가 아니거나, 존재하지 않는 디렉토리인 경우, 에러 처리 후 프롬프트 출력

✓ ssu_sdup 명령어 실행 시 출력은 다음과 같음
- 지정한 디렉토리 내에 중복 파일이 있는 경우, 각 중복 파일 리스트를 구분하여 한 세트씩 중복 파일 리스트 출력
  - 첫째 줄에는 각 중복 파일 리스트의 세트 번호와 파일 크기(바이트 단위), 해시값을 출력. 예시 7 참고
    ☞ 파일 크기는 천 단위마다 ','로 구분해야 함
  - 다음 줄부터, 중복 파일 리스트 내의 인덱스 번호와 파일의 절대 경로, 마지막 수정 시간, 마지막 접근 시간을 출력함. 예시 7 참고
    ☞ 세트 내 리스트 파일들의 출력순서는 파일 절대 경로가 짧은 것부터, 동일한 절대 경로는 아스키 코드 순서
    ☞ 마지막 수정 시간과 접근 시간은 stat 구조체 이용
  - 위 과정을 반복하여 전체 중복 파일 리스트 출력함
- 지정한 디렉토리 내에 중복 파일이 없는 경우, "No duplicates in (TARGET_DIRECTORY의 절대경로)" 출력 후 프롬프트 출력. 예시 10 참고
- 중복 파일 탐색 및 출력을 마치면 마지막 줄에 탐색 소요 시간 출력. 예시 7 참고
- 중복 파일 리스트가 존재하는 경우 "〉〉"를 출력하고 사용자 입력 대기

(예시 7). 파일 탐색 결과 출력 - 지정한 디렉토리 내 모든 중복 파일 리스트 출력

```
su@su-virtual-machine:~/2022-1_LSP/P2$ ./ssu_sdup
20220000> fmd5 * ~ ~ ~/test
---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/aaa/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/aa/a/bbb/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)

---- Identical files #2 (10 bytes - a61b545ae32677047a21014d7af49fee) ----
[1] /home/su/test/1 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:46:54)
[2] /home/su/test/2 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:47:09)
[3] /home/su/test/a b (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:03)

---- Identical files #3 (16 bytes - 7109c5edcae94719a22fb9f4023b32b0) ----
[1] /home/su/test/out3.bin (mtime : 2022-01-10 10:53:28) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/out4.bin (mtime : 2022-01-10 10:53:28) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/b/out3.bin (mtime : 2022-01-10 10:53:28) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/b/out4.bin (mtime : 2022-01-10 10:53:28) (atime : 2022-03-23 16:54:11)

---- Identical files #4 (32 bytes - 79e8cbdab7a957d55f52fa6da06241be) ----
[1] /home/su/test/a.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/c.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)

---- Identical files #5 (32 bytes - b82aa750b2124c0b00668a267a5dc3bf) ----
[1] /home/su/test/a/different_a.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/aa/different_b.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/b/different_c.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)

---- Identical files #6 (1,219,664 bytes - 6596d99856f98ba9476b455bde8f3ca5) ----
[1] /home/su/test/sample1.o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-22 17:31:45)
[2] /home/su/test/sample2.o (mtime : 2022-02-08 17:35:40) (atime : 2022-03-22 17:31:45)
[3] /home/su/test/test1 (copy).o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-23 15:38:19)
[4] /home/su/test/test1.o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-22 17:31:45)

---- Identical files #7 (1,885,256 bytes - 40b1c990a700fd241db3f09a402ff216) ----
[1] /home/su/test/example1.o (mtime : 2022-02-08 17:36:54) (atime : 2022-03-22 17:31:45)
[2] /home/su/test/example2.o (mtime : 2022-02-08 17:35:05) (atime : 2022-03-22 17:31:45)
[3] /home/su/test/example3.o (mtime : 2022-02-08 17:35:43) (atime : 2022-03-22 17:31:45)

Searching time: 0:018930(sec:usec)

>>
```

(예시 8). 파일 탐색 결과 출력 - 지정한 디렉토리 내 png 확장자 파일 중 중복 파일 리스트 출력

```
su@su-virtual-machine:~/2022-1_LSP/P2$ ./ssu_sdup
20220000> fmd5 *.png ~ ~ ~/test
---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/aaa/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/aa/a/bbb/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)

Searching time: 0:000259(sec:usec)

>>
```

(예시 9). 파일 탐색 결과 출력 - 지정한 디렉토리 내 32바이트 이상 1.8MB 이하 중복 파일 리스트 출력

```
20220000> fmd5 * 32 1.8MB ~/test
---- Identical files #1 (32 bytes - 79e8cbdab7a957d55f52fa6da06241be) ----
[1] /home/su/test/a.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/c.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)

---- Identical files #2 (32 bytes - b82aa750b2124c0b00668a267a5dc3bf) ----
[1] /home/su/test/a/different_a.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/aa/different_b.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/b/different_c.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)

---- Identical files #3 (1,219,664 bytes - 6596d99856f98ba9476b455bde8f3ca5) ----
[1] /home/su/test/sample1.o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-22 17:31:45)
[2] /home/su/test/sample2.o (mtime : 2022-02-08 17:35:40) (atime : 2022-03-22 17:31:45)
[3] /home/su/test/test1 (copy).o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-23 15:38:19)
[4] /home/su/test/test1.o (mtime : 2022-02-08 17:36:52) (atime : 2022-03-22 17:31:45)

Searching time: 0:010421(sec:usec)

>>
```

(예시 10). 파일 탐색 결과 출력 – 지정한 디렉토리 내 2MB 이상 파일 중 중복 파일이 없는 경우

```
20220000> fmd5 * 2mb ~ ~/test
No duplicates in /home/su/test
Searching time: 0:000148(sec:usec)

20220000>
```

✓ 〉〉 [SET_INDEX] [OPTION ...]
- 사용자가 선택한 [SET_INDEX]에 해당하는 중복 파일 리스트에서 [OPTION]에 따라 중복 파일 삭제
- 파일 삭제 후, 남아있는 중복 파일 리스트를 보여주고 "〉〉 "를 출력한 후 사용자 입력 대기
- 남아있는 중복 파일 리스트가 없으면 프롬프트 출력
- "exit"을 입력하면 "〉〉 Back to Prompt"를 출력하고 프롬프트 출력

(예시 11). 〉〉 exit 수행 결과

```
20220000〉 fmd5 * ~ 5 ~/test
---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/aaa/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/aa/a/bbb/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)

Searching time: 0:000218(sec:usec)

〉〉 exit
〉〉 Back to Prompt
20220000〉
```

- **[SET_INDEX]**
  - 전체 중복 파일 리스트에서 삭제하고자 하는 중복 파일 리스트의 세트 번호
  - [SET_INDEX] 인자에 입력이 없거나 범위를 벗어난 경우, 에러 처리 후 다시 입력 대기
- **[OPTION]**
  - 선택한 세트에서 파일을 삭제하기 위한 옵션
  - d, i, f, t 옵션이 존재하며, 이외의 옵션이 들어오면 에러 처리 후 "〉〉 " 출력하고 사용자 입력 대기
  - 옵션은 중복으로 사용하지 않음
  - d [LIST_IDX] : 선택한 세트에서 [LIST_IDX]에 해당하는 파일 삭제
    - ☞ [LIST_IDX] 인자에 입력이 없거나 범위를 벗어난 경우, 에러 처리 후 "〉〉 " 출력하고 사용자 입력 대기
    - ☞ 삭제 후 예시 12와 와 같이 삭제한 파일의 절대 경로 출력

(예시 12). d 옵션 출력

```
20220000〉 fmd5 * ~ 9 ~/test
---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/aaa/a.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/aa/a/bbb/a.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 16:54:11)

Searching time: 0:000213(sec:usec)

〉〉 1 d 3
"/home/su/test/aa/a/aaa/a.png" has been deleted in #1

---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/bbb/a.png (mtime : 2022-03-15 01:20:08) (atime : 2022-03-23 16:54:11)

〉〉
```

- i : 선택한 세트의 중복 파일 리스트에 있는 파일의 절대 경로를 하나씩 보여주면서 삭제 여부 확인 후 파일 삭제 또는 유지
  - ☞ 중복 파일 리스트 원소의 절대경로를 하나씩 보여주면서 삭제 여부 확인 메시지 출력
  - ☞ "y"나 "Y" 입력 시, 해당 파일 삭제, "n"나 "N" 입력 시, 해당 파일 유지
    - 이외의 입력이 들어오면 예외 처리 후 "〉〉 " 출력하고 사용자 입력 대기
  - ☞ 모든 파일에 대해 삭제 여부 확인 완료하면 전체 중복 파일 리스트 출력
    - 선택한 세트의 중복 파일 리스트에 있는 파일을 모두 삭제하거나 하나만 남는 경우 중복 파일 리스트에서 제거됨

(예시 13). i 옵션 출력

```
20220000〉 fmd5 * 10 15 ~/test
---- Identical files #1 (10 bytes - a61b545ae32677047a21014d7af49fee) ----
[1] /home/su/test/1 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:46:54)
[2] /home/su/test/2 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:47:09)
[3] /home/su/test/a b (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:03)
[4] /home/su/test/a b (another copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)
[5] /home/su/test/a b (copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)

Searching time: 0:000226(sec:usec)

〉〉 1 i
Delete "/home/su/test/1"? [y/n] y
Delete "/home/su/test/2"? [y/n] N
Delete "/home/su/test/a b"? [y/n] Y
Delete "/home/su/test/a b (another copy)"? [y/n] n
Delete "/home/su/test/a b (copy)"? [y/n] y

---- Identical files #1 (10 bytes - a61b545ae32677047a21014d7af49fee) ----
[1] /home/su/test/2 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:47:09)
[2] /home/su/test/a b (another copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)

〉〉 1 i
Delete "/home/su/test/2"? [y/n] y
Delete "/home/su/test/a b (another copy)"? [y/n] y

20220000〉
```

- f : 가장 최근에 수정한 파일을 남겨두고 나머지 중복 파일을 삭제
  - ☞ 삭제 후 예시 14와 같이 가장 최근에 수정한 파일의 절대 경로와 수정 시간 출력
  - ☞ f 옵션 사용 시, 파일이 하나만 남기 때문에 중복 파일 리스트에서는 제거됨

(예시 14). f 옵션 출력

```
20220000> fmd5 * 32 1kb ~/test
---- Identical files #1 (32 bytes - 79e8cbdab7a957d55f52fa6da06241be) ----
[1] /home/su/test/a.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/c.bin (mtime : 2022-01-19 12:42:25) (atime : 2022-03-23 16:54:11)

---- Identical files #2 (32 bytes - b82aa750b2124c0b00668a267a5dc3bf) ----
[1] /home/su/test/a/different_a.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/aa/different_b.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/b/different_c.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)

Searching time: 0:000373(sec:usec)

>> 1 f
Left file in #1 : /home/su/test/a.bin (2022-01-19 12:42:25)

---- Identical files #1 (32 bytes - b82aa750b2124c0b00668a267a5dc3bf) ----
[1] /home/su/test/a/different_a.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/aa/different_b.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)
[3] /home/su/test/b/different_c.bin (mtime : 2022-01-19 22:30:12) (atime : 2022-03-23 16:54:11)

>>
```

- t : 가장 최근에 수정한 파일을 남겨두고 나머지 중복 파일을 휴지통으로 이동
  - ☞ 삭제 후 예시 15와 같이 가장 최근에 수정한 파일의 절대 경로와 수정 시간 출력
  - ☞ t 옵션 사용 시, 파일이 하나만 남기 때문에 중복 파일 리스트에서는 제거됨

(예시 15). t 옵션 출력

```
20220000〉 fmd5 * ~ 10 ~/test
---- Identical files #1 (5 bytes - 2b00042f7481c7b056c4b410d28f33cf) ----
[1] /home/su/test/aa/b/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[2] /home/su/test/b/b/ccc.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 17:11:30)
[3] /home/su/test/aa/a/aaa/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)
[4] /home/su/test/aa/a/bbb/a.png (mtime : 2022-01-15 01:20:08) (atime : 2022-03-23 16:54:11)

---- Identical files #2 (10 bytes - a61b545ae32677047a21014d7af49fee) ----
[1] /home/su/test/1 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:46:54)
[2] /home/su/test/2 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:47:09)
[3] /home/su/test/a b (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:03)
[4] /home/su/test/a b (another copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)
[5] /home/su/test/a b (copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)

Searching time: 0:000249(sec:usec)

〉〉 1 t
All files in #1 have moved to Trash except "/home/su/test/aa/b/a.png" (2022-01-15 01:20:08)

---- Identical files #1 (10 bytes - a61b545ae32677047a21014d7af49fee) ----
[1] /home/su/test/1 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:46:54)
[2] /home/su/test/2 (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:47:09)
[3] /home/su/test/a b (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:03)
[4] /home/su/test/a b (another copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)
[5] /home/su/test/a b (copy) (mtime : 2022-01-23 15:37:54) (atime : 2022-03-23 15:38:19)

〉〉
```

- 나) 내장명령어 2, fsha1
  - ✓ 사용법 : fsha1 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY]
    - SHA1 해시값을 구하기 위해 SHA1(openssl/sha.h)를 사용
      - ☞ Linux, Ubuntu :"sudo apt-get install libssl-dev"로 라이브러리 설치 필요
      - ☞ Linux, Fedora :"sudo dnf-get install libssl-devel"로 라이브러리 설치 필요
      - ☞ MacOS : homebrew (https://brew.sh/ 참고) 설치 -〉 % brew install openssl
    - SHA1 관련 함수를 사용하기 위해 컴파일 시 "-lcrypto" 옵션 필요
    - SHA1() 사용법은 https://www.openssl.org/docs/man1.1.1/man3/SHA1.html 및 https://github.com/Chronic-Dev/openssl/blob/master/crypto/sha/sha1.c 참고
  - ✓ 내장명령어 1, fmd5의 모든 실행결과와 동일한 형태로 출력(본 명세에서는 예시는 따로 보여주지 않음)

- 다) 내장명령어 3, exit
  - ✓ 프로그램을 종료시키는 명령어
  - ✓ exit 명령어 실행 시 명령어를 실행시키는 프로세스, ssu_sdup 프로세스 모두가 종료되어야 함

(예시 16). exit 명령어 출력 예시

```
su@su-virtual-machine:~/2022-1_LSP/P2$ ./ssu_sdup
20220000> exit
Prompt End
su@su-virtual-machine:~/2022-1_LSP/P2$
```

- 라) 내장명령어 4, help
  - ✓ 명령어 사용법을 출력하는 명령어

(예시 17). help 명령어 출력 예시

```
su@su-virtual-machine:~/2022-1_LSP/P2$ ./ssu_sdup
20220000> help
Usage:
  > fmd5/fsha1 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY]
     >> [SET_INDEX] [OPTION ... ]
        [OPTION ... ]
        d [LIST_IDX] : delete [LIST_IDX] file
        i : ask for confirmation before delete
        f : force delete except the recently modified file
        t : force move to Trash except the recently modified file
  > help
  > exit

20220000>
```

○ make와 Makefile

- make : 프로젝트 관리 유틸리티
  - ✓ 파일에 대한 반복 명령어를 자동화하고 수정된 소스 파일만 체크하여 재컴파일 후 종속된 부분만 재링크함
  - ✓ Makefile(규칙을 기술한 파일)에 기술된 대로 컴파일 명령 또는 쉘 명령을 순차적으로 실행함
- Makefile의 구성
  - ✓ Macro(매크로) : 자주 사용되는 문자열 또는 변수 정의 (컴파일러, 링크 옵션, 플래그 등)
  - ✓ Target(타겟) : 생성할 파일
  - ✓ Dependency(종속 항목) : 타겟을 만들기 위해 필요한 파일의 목록
  - ✓ Command(명령) : 타겟을 만들기 위해 필요한 명령(shell)

```
Macro

Target : Dependency1 Dependency2 ...
<-Tab->Command 1
<-Tab->Command 2
<-Tab->...
```

- Makefile의 동작 순서
  - ✓ make 사용 시 타겟을 지정하지 않으면 제일 처음의 타겟을 수행
  - ✓ 타겟과 종속 항목들은 관습적으로 파일명을 명시
  - ✓ 명령 항목들이 충족되었을 때 타겟을 생성하기 위해 명령 (command 라인의 맨 위부터 순차적으로 수행)
  - ✓ 종속 항목의 마지막 수정 시간(st_mtime)을 비교 후 수행
- Makefile 작성 시 참고사항
  - ✓ 명령의 시작은 반드시 Tab으로 시작해야함
  - ✓ 한 줄 주석은 #, 여러 줄 주석은 ₩
  - ✓ 자동 매크로 : 현재 타겟의 이름이나 종속 파일을 표현하는 매크로

매크로	설명
$?	타겟보다 최근에 변경된 종속 항목 리스트 (확장자 규칙에서 사용 불가능)
$^	현재 타겟의 종속 항목 (확장자 규칙에서 사용 불가능)
$<	타겟보다 최근에 변경된 종속 항목 리스트 (확장자 규칙에서만 사용 가능)
$*	타겟보다 최근에 변경된 종속 항목 리스트 (확장자 규칙에서만 사용 가능)
$@	현재 타겟의 이름

- Makefile 작성 예시

(예시 18). Makefile

```
test : test.o add.o sub.o
    gcc test.o add.o sub.o -o test

test.o: test.c
    gcc -c test.c

add.o: add.c
    gcc -c add.c

sub.o: sub.c
    gcc -c sub.c

clean :
    rm test.o
    rm add.o
    rm sub.o
    rm test
```

(예시 19). 매크로를 사용한 경우

```
OBJECTS = test.o add.o sub.o
TARGET = test
CC = gcc

$(TARGET) : $(OBJECTS)
    $(CC) -o $(TARGET) $(OBJECTS)

test.o: test.c
    $(CC) -c test.c

add.o: add.c
    $(CC) -c add.c

sub.o: sub.c
    $(CC) -c sub.c
```

(예시 20). 자동 매크로를 사용한 경우

```
OBJECTS = test.o add.o sub.o
TARGET = test
CC = gcc

$(TARGET) : $(OBJECTS)
    $(CC) -o $@ $^

test.o: test.c
    $(CC) -c $^

add.o: add.c
    $(CC) -c $^

sub.o: sub.c
    $(CC) -c $^
```

- (예시 21). Makefile 수행 예시

oslab@a-VirtualBox:~$ make gcc -c test.c gcc -c add.c gcc -c sub.c gcc test.o add.o sub.o –o test oslab@a-VirtualBox:~$	oslab@a-VirtualBox:~$ make clean rm test.o rm add.o rm sub.o rm test oslab@a-VirtualBox:~$

○ 과제 구현에 필요한 함수 (필수로 사용해야 하는 것은 아님)

- 1. memset : s가 가리키는 메모리 영역의 처음 n바이트를 상수 바이트 c로 채우는 함수

```
#include <string.h>
void *memset(void *s, int c, size_t n);

a pointer to the memory area s.
```

- 2. strtok : 특정 문자 기준으로 문자열을 분리하는 함수

```
#include <string.h>
char *strtok(char *restrict str, const char *restrict delim);

 return a pointer to the next token, or NULL if there are no more tokens.
```

- 3. strstr : 문자열에서 부분 문자열의 위치를 찾는 함수

```
#include <string.h>
char *strstr(const char *haystack, const char *needle);

return a pointer to the beginning of the located substring, or NULL if the substring is not found. If needle is the empty string, the return value is always haystack itself.
```

○ 보고서 제출 시 유의 사항

- 보고서 제출 마감은 4월 20일 11:59 PM
- 지연 제출 시 감점 : 1일 지연 시 마다 30% 감점, 3일 지연 후부터는 미제출 처리
- 압축 오류, 파일 누락 관련 감점 등은 강의계획서 참고

○ 프로그램 구현 시 유의 사항 및 감점

- ssu_sdup의 모든 출력 내용에 대해서 출력 형식 미 준수 시 감점(공백 또는 탭은 제외) 50% 감점

○ 구현 배점

- 가) 프롬프트 출력 [5점]
- 나) 내장명령어 1, fmd5 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY], 아래 기타 1을 제외한 기본 기능 [20점]
- 나) 내장명령어 2, fsha1 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY], 아래 기타 1을 제외한 기본 기능 [20점]
- 다) 내장명령어 3, exit [5점]
- 라) 내장명령어 4, help [5점]
- 기타1) fmd5 내장명령어, 〉〉 [SET_IDX] [OPTION ...]에서 4가지 옵션 처리 [20점]
- 기타2) fsha1 내장명령어, 〉〉 [SET_IDX] [OPTION ...]에서 4가지 옵션 처리 [20점]
- 기타3) gettimeofday() 로 탐색시간 측정 [5점]
- 추가기능) 추가로 학생이 스스로 옵션을 추가할 경우 옵션당 3점, 최대 10점까지 부여 가능. 옵션을 추가할 경우 내장명령어 help에도 설명이 들어가야 하며, 추가한 옵션 기능을 상세하게 설명하고, 예시를 넣어 테스트한 내용을 보고서에 포함시켜야 함

○ 필수 구현 사항

- 가) 프롬프트 출력
- 나) 내장명령어 1, fmd5 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY], 〉〉 [SET_IDX] [OPTION ...] 4가지 옵션을 제외한 기본 기능
- 다) 내장명령어 2, fsha1 [FILE_EXTENSION] [MINSIZE] [MAXSIZE] [TARGET_DIRECTORY], 〉〉 [SET_IDX] [OPTION ...] 4가지 옵션을 제외한 기본 기능
- 다) 내장명령어 3, exit
- 라) 내장명령어 4, help
- 기타1) fmd5 내장명령어, 〉〉 [SET_IDX] [OPTION ...]에서 d, t 옵션 처리
- 기타2) fsha1 내장명령어, 〉〉 [SET_IDX] [OPTION ...]에서 d, t 옵션 처리
- 기타3) gettimeofday() 로 탐색시간 측정