시스템프로그래밍실습 보고서

Assignment 1-2

과 목 시스템프로그래밍실습

담당교수 이기훈교수님

학 과 컴퓨터정보공학부

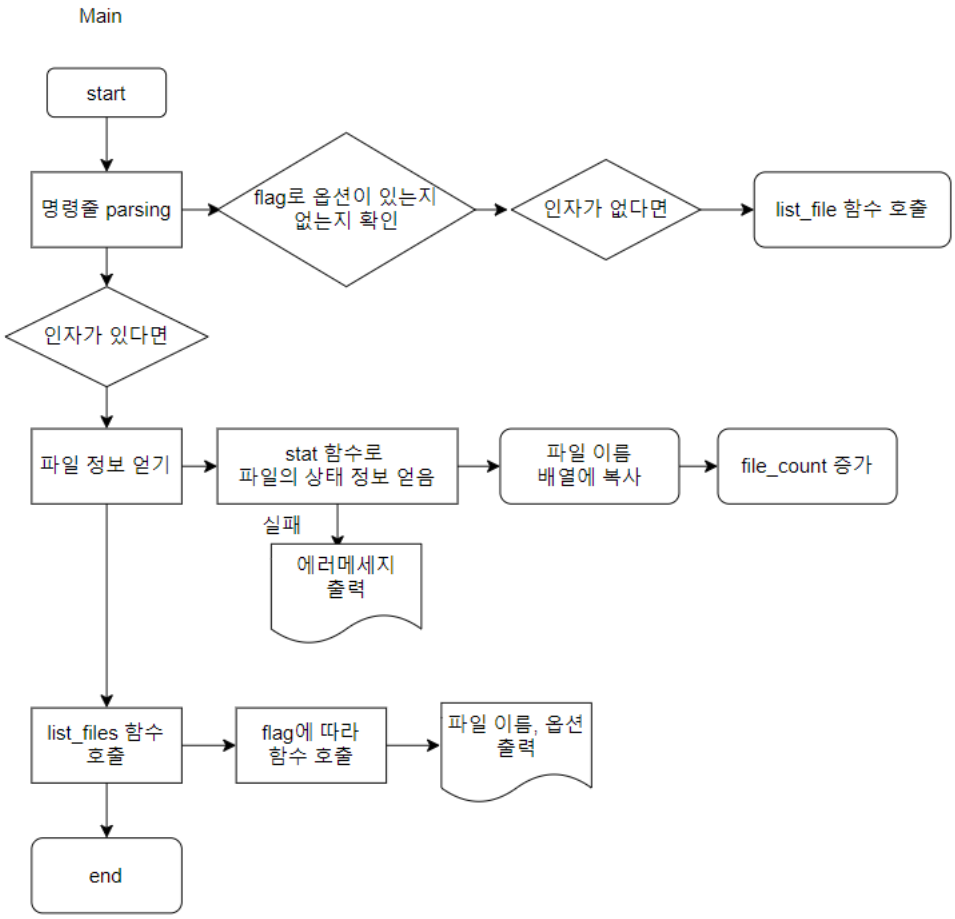
학 번 2021202058

이 름 송채영

1. Introduction

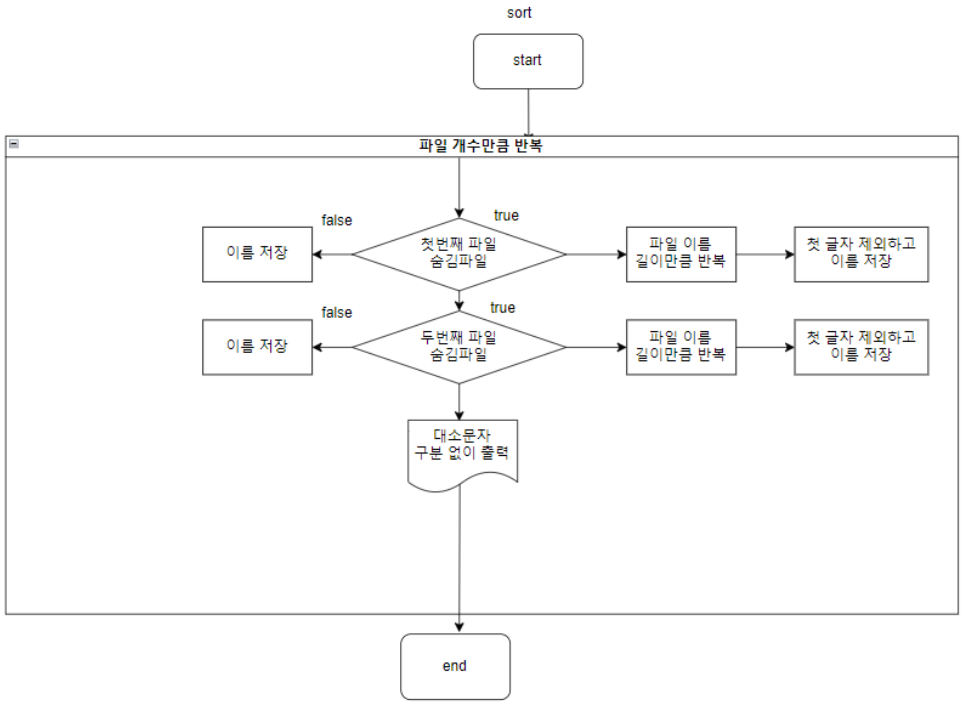
ls 명령어와 ls 명령어의 옵션(-a, -l, -al)을 직접 구현해본다. 구현 과정에서 linux 의 data type 에 대해 알아보고 이를 활용한다. 지난 시간에 활용한 dir, struct dirent 이외에도, struct passwd, struct group, struct stat, struct tm 등을 사용해 파일의 세부정보를 출력한다. 현재 작업 디렉터리를 가져오는 system call 에 대해 알아보며 이를 코드에 적용한다.

2. Flow chart

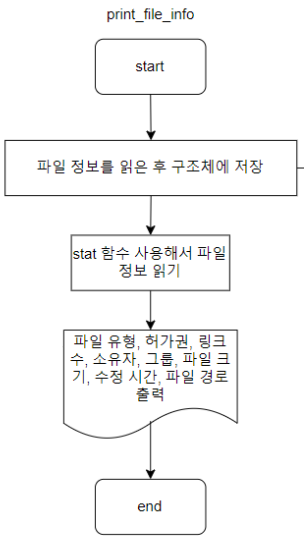


코드의 전체적인 흐름을 flow chart 로 나타내었다. main 함수에 대해 자세히 설명해보면 다음과 같다. 프로그램이 시작하면 getopt 함수를 사용해 명령줄 인자를 parsing 한다. getopt 함수는 argv 배열과 인자의 개수 argc, 그리고 옵션 문자열을 입력으로 받아 옵션 문자 하나를 읽어온다. 옵션이 있는지 없는지는 flag 로 판단하며 만약 인자가 없다면 list_file 함수를 호출한다. 인자가 있다면 파일 정보를 얻는다. 이때 stat 함수로 파일의

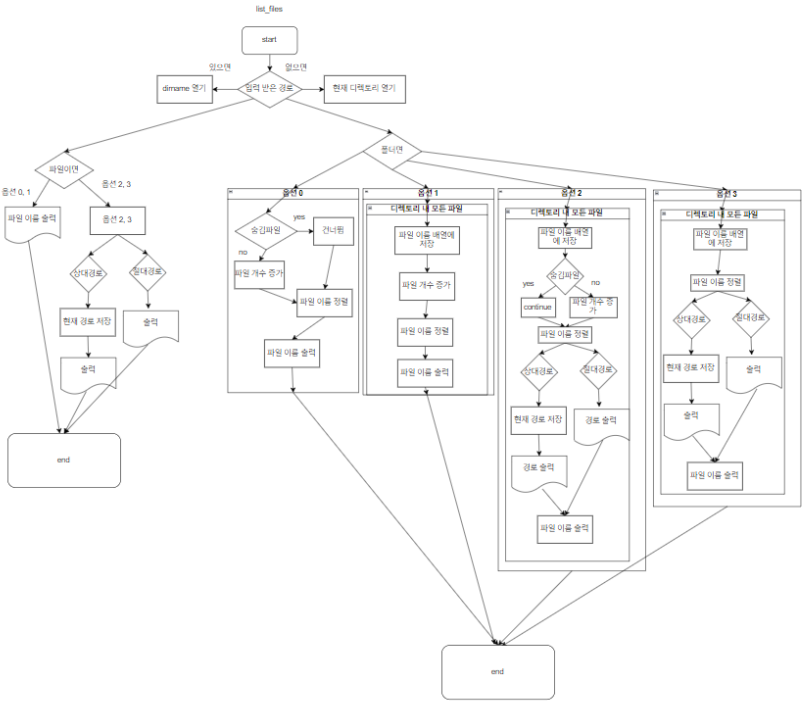
상태 정보를 얻는데, 정보를 얻는데 실패하면 에러 메세지를 출력하며, 아닐 경우 파일 이름을 배열에 복사한 후 file_count 를 증가시킨다. 그 후 list_file 함수를 호출하는데 flag 에 따라 함수를 호출해 파일 이름과 옵션을 출력한다. 프로그램을 종료한다.



다음은 sort 함수의 flow chart 이다. 우선 파일의 개수만큼 반복한다. 첫 번째 파일이 숨김파일일 경우, 파일의 이름을 길이만큼 반복하고 첫 글자를 제외한 이름을 저장한다. 숨김파일이 아닐 경우 이름을 저장해준다. 두 번째 파일도 동일하게 진행한다. 그 후 대소문자 구분 없이 출력해준 후 프로그램을 종료한다.



다음은 print_file_info, 파일 정보를 출력하는 함수의 flow chart 이다. 파일 정보를 읽어 구조체 변수인 filestat 에 저장한다. stat 함수를 사용해서 파일 정보를 읽은 후 파일 유형, 허가권, 링크 수 , 소유자, 그룹, 파일 크기, 수정시간, 파일 경로를 출력한 후 프로그램을 종료한다.



다음은 list_file, 디렉토리 내의 파일 목록을 출력하는 함수이다. 입력 받은 경로가 없으면 현재 디렉토리를 열고, 있으면 dirname 을 연다. dirname 이 파일일 때와 폴더일때로 나눈다. 먼저 파일일 때 옵션 0, 1 즉 옵션이 없을 때와 -a 일때는 파일 이름을 출력한다. 옵션이 2, 3 즉 -l 또는 -al 일 때 상대경로와 절대경로를 나누어 출력한다. 상대경로일때 현재 경로를 저장하고 출력하며, 절대 경로는 바로 출력한다. 폴더일때는 옵션 0, 1, 2, 3 일때로 나누는데, 옵션 0 일땐, 숨김파일이면 건너뛰고 숨김파일이 아닐경우 파일 개수를 증가한다. 후에 파일 이름을 정렬하고 파일 이름을 출력한다. 옵션 1 일땐, 디렉토리 내 모든 파일을 순회하며, 파일 이름을 배열에 저장한 후 파일 개수를 증가시키고 파일 이름을 정렬하고 파일 이름을 출력한다. 옵션 2 일땐, 디렉터리 내 모든 파일을 순회하며, 파일 이름을 배열에 저장한 후 숨긴 파일이면 넘어가고 아닐 경우 파일 개수를 증가한다. 그 후 파일 이름을 정렬하고 상대경로와 절대경로로 나눈다. 그 후 파일 이름을 출력해준다. 마지막으로 옵션 3 일땐, 파일 이름을 배열에 저장한 후 파일 이름을 정렬한다. 상대경로와 절대경로로 나눈 후 파일 이름을 출력해준다.

3. Pseudo Code

```
sort(파일이름, 파일 수)
{
	for(파일 수만큼 파일 이차원 배열 순회)
	{
		if(첫번째 파일 첫 글자가 숨김 파일이면)
		{
			filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
		}
		else(숨김파일이 아니면)
		{
			이름 저장
		}
		if(두번째 파일 첫 글자가 숨길 파일이면)
		{
			filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
		}
		else(숨김파일이 아니면)
		{
			이름 저장
		}
		대소문자 구분없이 비교
		}
	}
}
```

```
파일 정보 출력(파일 경로) //print_file_info
{
	파일 정보 읽어 구조체 변수 filestat에 저장
	파일 유형, 허가권, 링크 수 , 소유자, 그룹, 파일 크기, 수정 시간 출력
}
```

```
파일 정보 출력(파일 경로) //print_file_info
{
	파일 정보 읽어 구조체 변수 filestat에 저장
	파일 유형, 허가권, 링크 수 , 소유자, 그룹, 파일 크기, 수정 시간 출력
}

파일 목록 출력(dirname, option) //list_files
{
	if(입력 받은 경로가 없으면)
	{
		현재 디렉토리 열기
	}
	else(있으면)
	{
		 인자로 입력받은 디렉토리 열기
	}

	if(파일이면)
	{
		if(옵션이 없거나 -a 옵션일때)
		{
			파일 이름 출력
		}
		else(옵션이 -l이거나 -al 일때)
		{
			if(절대경로)
			{
				현재 경로 출력
			}
			else(상대경로)
			{
				현재 경로 저장 후 출력
			}
		}
```

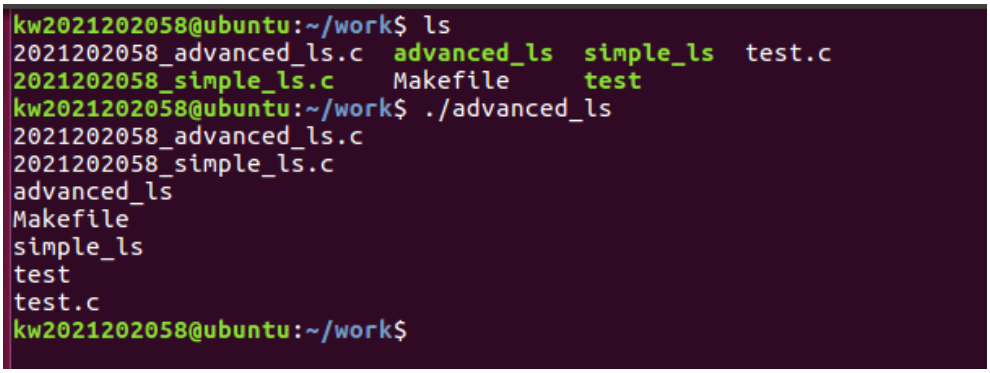
```
        파일 정보 구조체에 저장 후 파일 정보 출력
}

//directory
if(옵션이 없을 때)
{
        while(모든 파일 순회)
        {
                파일 이름 배열에 저장
                숨김파일이면 건너뛰고 아니면 파일 개수 증가
        }
        파일 이름 정렬 후 파일 이름 출력
}
else if(-a옵션 일때)
{
        while(모든 파일 순회)
        {
                파일 이름 배열에 저장 후 파일 개수 증가
        }
        파일 이름 정렬 후 파일 이름 출력
}
else if(-l 옵션일때)
{
        while(모든 파일 순회)
        {
                파일 이름 배열에 저장
                숨김파일이면 건너뛰고 아니면 파일 개수 증가
        }
        파일 이름 정렬
        if(절대경로)
        {
                현재 경로 출력
        }
```

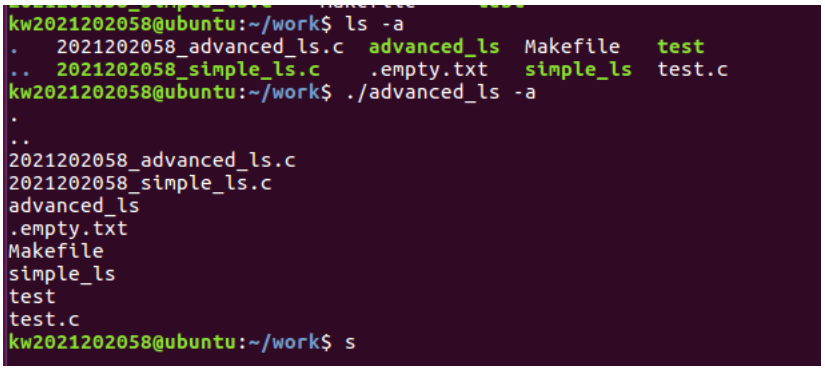
```
                else(상대경로)
                {
                        현재 경로 저장 후 출력
                }
                파일 정보 출력
        }
        else(-al 옵션일때)
        {
                while(모든 파일 순회)
                {
                        파일 이름 배열에 저장 후 파일 개수 증가
                }
                파일 이름 정렬
                if(절대경로)
                {
                        현재 경로 출력
                }
                else(상대경로)
                {
                        현재 경로 저장 후 출력
                }
                파일 정보 출력
        }
        디렉토리 닫기
}
```

```
main(argc, argv)
{
      명령줄 parsing
      {
            a 옵션일 때 aflag +1
            l 옵션일 때 lflag +1
            알 수 없는 옵션일 때 에러메세지 출력
      }
      if(인자가 없을 때)
      {
            옵션에 따라 list_files 함수 호출
      }
      else(인자가 있을 때)
      {
            if(인자로 주어진 파일 반복)
            {
                  if 파일 정보 얻는데 실패 시 오류 메세지 출력
                  else 성공했을 때 파일 이름 배열에 복사 후 파일 개수 증가
            }
      }
      for(파일 개수만큼)
      {
            aflag와 lflag에 따라 list_files 함수 호출
      }
```

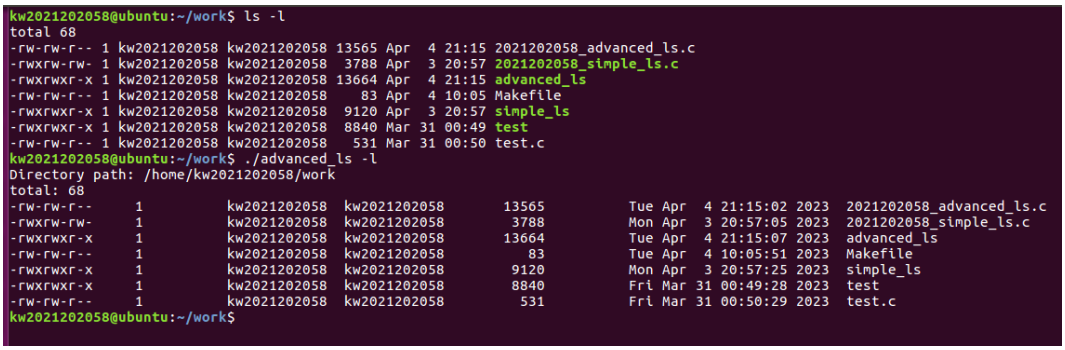
4. 결과화면



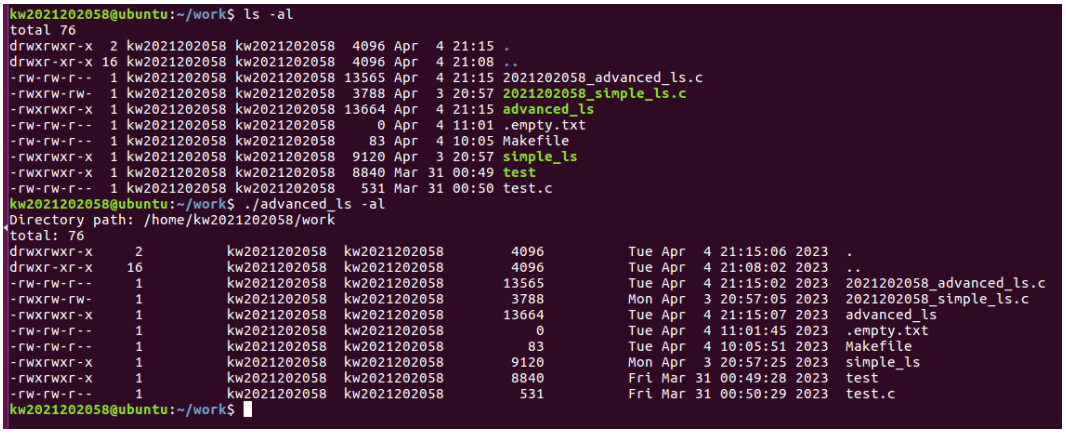
ls 명령어와 구현한 ls 명령어를 출력한 사진이다. ls 명령어를 사용했을 때와 동일하게 출력되며, 정렬 역시 잘 된 것을 확인할 수 있다.



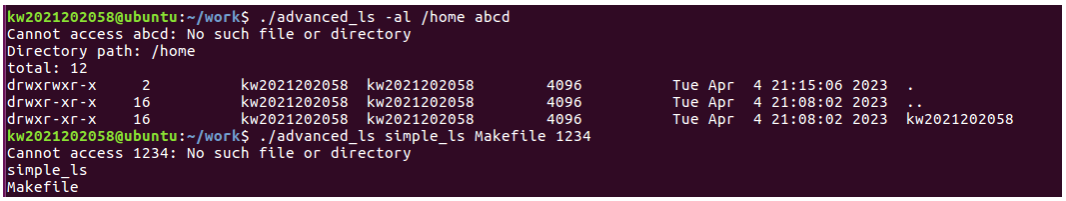
ls -a 옵션과 구현한 -a 옵션을 출력한 사진으로 ls -a 옵션을 사용했을 때와 동일하게 출력된다. 숨김 파일이 출력되며 정렬 역시 처음 '.'을 제외한 문자열이 동일한 조건으로 정렬되어 있는 것을 확인할 수 있다.

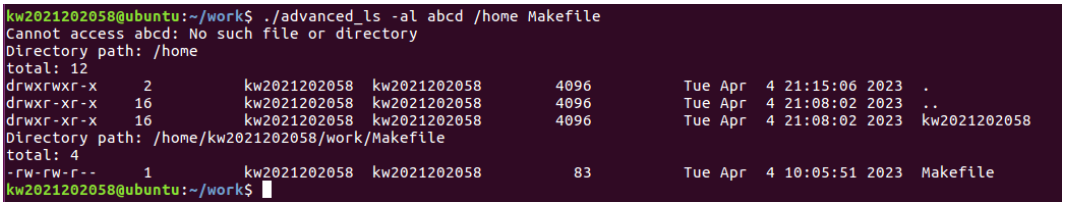


ls -l 옵션과 구현한 -l 옵션을 출력한 사진으로 ls -l 옵션을 사용했을 때와 동일하게 출력된다. 파일 유형, 허가권, 링크 수, 소유자, 그룹, 파일 크기, 수정 시간 등 출력 포맷이 동일하며 정렬되어 출력된 것을 확인할 수 있다. 또한 directory path 와 total 을 출력하는 조건 역시 만족한 것을 확인할 수 있다.



ls -al 옵션과 구현한 -al 옵션을 출력한 사진으로 ls -al 옵션을 사용했을 때와 동일하게 출력된다. 파일 유형, 허가권, 링크 수, 소유자, 그룹, 파일 크기, 수정 시간 등 출력 포맷이 동일하며 숨김 파일 역시 출력되는 것을 볼 수 있다. 또한 정렬 역시 처음 '.'을 제외한 문자열이 동일한 조건으로 정렬되어 출력된 것을 확인할 수 있으며, directory path 와 total 을 출력하는 조건 역시 만족한 것을 확인할 수 있다.





과제에서 요구한 추가 조건인 directory path 와 1k block 의 수, 즉 total 을 출력해 주었으며 존재하지 않는 파일이나 디렉토리가 입력으로 들어온 경우를 보여주는 결과화면이다. 먼저 존재하지 않는 파일 및 디렉토리에 관한 에러 메시지는 상단에 출력되도록 하는 조건을 만족하였고, 올바른 파일이나 디렉토리는 하단에 입력한 순서대로 출력되도록 하는 조건 역시 만족하였다. 또한 상대경로와 절대경로에 대해서도 출력을 잘 하는 것을 확인할 수 있다.

5. 고찰

이번 과제를 수행하면서 예외처리를 하는 것과 segmentation fault 가 떴을 때 그것을 해결하는 것이 가장 어려웠던 것 같다. makefile 을 통한 컴파일에서도 세그먼트 오류는 나타나지 않기 때문에 코드를 하나하나 살펴보고 코드의 흐름에 따라 값을 넣어보고

오류가 나는 곳을 찾아 해결했던 것 같다. 완벽하다고 생각했던 부분에서 초기화를 해주지 않아 쓰레기 값이 들어가는 등 생각보다 간단하고 사소한 실수로 세그먼트 오류가 났다. 오류가 나는 곳을 찾기 위해서 printf 를 활용하여 어디에 값이 잘못 들어가고 어디서 출력이 안 되는지를 확인해 주었다. 또한 절대경로와 상대경로를 넣어주기 전에는 잘 출력되던 코드가 경로를 넣어준 후 출력되지 않았고, stat 에 0 이 들어가 출력이 되지 않는다고 생각하여 파일 경로가 NULL 일 때라는 조건을 추가해주어 해결하였다. 옵션을 어떻게 구현해야 할지 생각이 많았지만, 시스템프로그래밍 실습 시간에 배운 ppt 를 활용하여 옵션에 따라 해당 명령어를 수행하도록 코드를 구현했다. linux 에서의 C 와 window 에서의 C 가 조금씩 다른 것 같아 공부가 좀 더 필요하다고 느꼈다. 또한 지난 시간에 배웠던 내용들에 매 시간 새로운 함수와 헤더파일에 대해 추가로 배우며 한 코드에서 사용한다는 것이 흥미로웠지만 익숙해지는데 시간이 꽤 걸렸던 것 같다.

6. Reference

시스템프로그래밍실습 강의자료