**[506489] System Programming (F’18)**

**Term Project (Week 3, Final) Report**

# 팀 정보

|  |  |  |
| --- | --- | --- |
| No. | 이름 | 학번 |
| 1 | 서종찬 | 20125141 |
| 2 | 안혁진 | 20135140 |
| 3 | 백은규 | 20167163 |
| 4 |  |  |
| 팀 이름: 9조 | | |

<목차>

[팀 정보 1](#_Toc532690680)

[결과물 제출 2](#_Toc532690681)

[보고서 작성 가이드 2](#_Toc532690682)

[Part I: 최종 보고서 3](#_Toc532690683)

[Part II: 데모/시연 시나리오 및 결과물 5](#_Toc532690684)

[Part III: 프로젝트 진행중 발생한 문제점 및 해결방안 9](#_Toc532690685)

[Part IV: 개선방향 10](#_Toc532690686)

\* 보고서 작성이 완료되면 <목차>를 업데이트 해 주세요. (방법: “목차” 클릭>”목차 업데이트”>”목차 전체 업데이트”)

# 결과물 제출

* SmartCampus에 업로드 해야 하는 제출물 목록은 다음과 같습니다.
  + Report(프로젝트 보고서), **PDF 형식**
  + 프로젝트 발표자료 (ppt, pptx, pdf 등)
  + GitHub 프로젝트를 다운로드 한 zip 파일
* 제출 기한을 넘기면 자동으로 0점 처리됩니다.
* 팀 과제이며, 팀별로 한 명만 제출하면 됩니다.

\* 보고서 및 발표자료도 GitHub에서 관리해야 합니다!!!

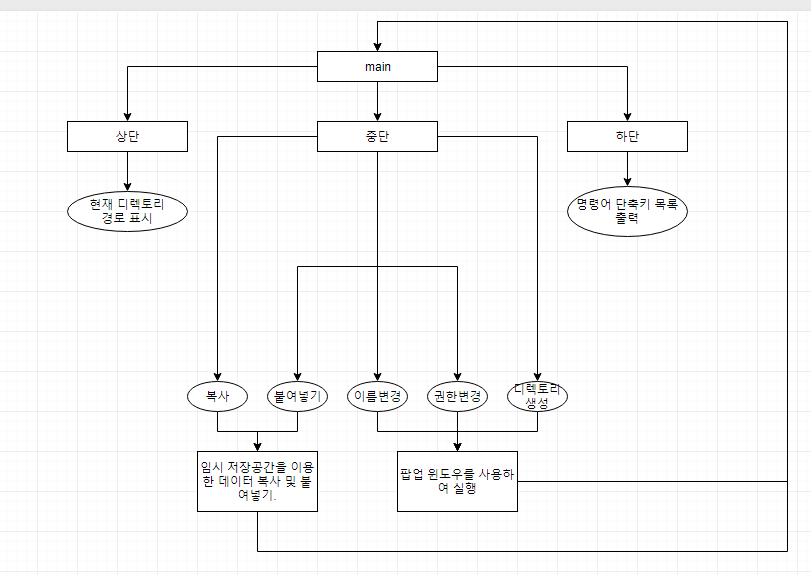
# 보고서 작성 가이드

* Week 3 시작: 2018. 12. 12. (Wed) 1:00pm
* Week 3 마감: 2018. 12. 17. (Mon) 12:59pm
* 매 주차별로 반드시 1회 이상 GitHub에 commit한 내역이 있어야 합니다. (팀별로 1회 이상, 개인별 1회 아님). GitHub에 commit한 내역이 없을 경우, 큰 감점을 받게 됩니다.

본 보고서는 총 3개의 파트로 구성되어 있습니다.

* Part I: 최종 보고서
  + 1주차 보고서 내용 전체를 포함하여 작성하고, 그 동안 추가/수정된 내용이 있다면 그에 맞게 보고서를 수정해야 합니다.
* Part II: 데모/시연 결과물
  + 개발한 프로그램이 정상적으로 구동된다는 것을 보여주세요.
  + 데모/시연을 위해 사용한 시나리오를 자세하게 설명하고, 해당 시나리오에 대한 단계별 데모 결과물(예: 스크린샷)을 첨부하거나 또는 데모 영상을 촬영해서 첨부파일로 제출해도 됩니다.
* Part III: 프로젝트 진행중 발생한 문제점 및 해결방안
  + 2주차 보고서 내용중 “프로젝트 진행 중 발생하는 문제점 및 해결방안”에 대한 내용을 포함하는 내용으로 작성하세요.
  + 그 외에 추가로 발생한 문제점이 있었다면 해당 문제점 및 해결방안을 기술하세요.
* Part IV: 개선방향
  + 개발 결과물을 앞으로 어떻게 개선할 수 있을지에 대한 의견을 서술하세요.

# Part I: 최종 보고서

* **프로젝트 제목**: 비주얼 쉘 탐색기
* **선정 동기**: 이번에 리눅스를 처음 사용해보았는데 명령어가 복잡하고 오타로 인한 오류도 자주 발생하는 것이 불편하여 좀 더 직관적이고 편리한 파일 관리자가 있으면 좋을 것 같다는 생각을 하여 비주얼 쉘 파일 탐색기를 주제로 제작하게 되었습니다.
* **프로젝트 내용**: 수업시간에 배운 Ncurses 라이브러리를 통한 GUI 환경을 제공.
* **프로젝트 초기 개발 목표**: 텍스트 편집기 nano와 비슷하게 창 아래 쪽에 단축키 목록을 보여주고 사용자로부터 입력을 받아 지정된 명령을 수행하게 하고 복사, 붙여넣기, 파일이동, 생성, 삭제, 권한 변경 등을 사용할 수 있게 한다.
* **프로젝트 개발/구현 내용**: 화면을 3등분하여 맨 위 블록에는 현재 디렉토리의 경로를 보여주며, 중간 블록 부분은 해당 디렉토리의 목록을 보여주고 맨 아래 블록에서는 단축키 명령어 목록을 보여준다.
* **어플리케이션 구상도:** 
* **기대효과 및 활용방안:** 리눅스 명령어를 많이 다루어 보지 못한 리눅스 초심자들을 위해 좀 더 직관적으로 명령어 사용을 도와주고 익숙해 보이는 인터페이스를 제공하고, 리눅스 관리자를 제외한 쉘 환경에서 파일 관리가 익숙하지 않은 사용자들이 좀 더 쉽게 파일을 관리할 수 있게 해준다.
* **프로젝트 최종 구현 내용:** 상단 블록 부분은 현재 보고 있는 디렉토리의 현위치를 표기해주고 중간 블록 부분은 선택된 디렉토리의 목록을 출력하고 개체가 디렉토리라면 “[디렉토리]” 일반 파일이라면 파일의 크기를 표기해준다. 하단 블록 부분은 초기에 구상했던 대로 단축키를 눌러 수행할 수 있는 명령어 목록을 보여준다. 하지만, 기존엔 탐색기 단축키로 복사, 붙여넣기, 이름변경, 권한변경, 디렉토리 생성 기능 명령을 구현하려 했으나 프로그램을 개발하는 난이도와 부족한 시간으로 인해 아쉽게도 디렉토리 생성 기능과 이름변경 기능 까지만 구현이 되었음.

# Part II: 데모/시연 시나리오 및 결과물

|  |
| --- |
| * **초기 실행 화면** |
|  |
| 🡪 키보드 방향키를 이용해 상하좌우로 이동이 가능하고 Enter를 눌렀을 때 선택된 개체가 디렉토리라면 선택한 디렉토리로 이동이 됨. 화면의 좌상단에 보이는 “ .. “ 을 선택하면 현재 디렉토리에서 상위 디렉토리로 이동함. 하단 블록 부분에 단축키가 있지만 현재 구동 가능한 단축키는 Mkdir과 Rename만 가능함. 프로그램 종료는 키보드 F12를 입력하면 종료된다. |

|  |
| --- |
| * **Rename 기능** |
| 🡪디렉토리 목록 중 “gggg.c” 파일에 커서를 두고 ctrl + n 버튼을 눌러 Rename 명령 시 |
| 🡪 새로 지정할 이름은 “newName.c” |
| 🡪 이름이 변경됨 |

|  |
| --- |
| * **Mkdir 기능** |
| 🡪 명령어 입력 후 생성할 디렉토리 이름은 “newDirectory”로 입력 후 확인 |
| 🡪 newDirectory가 생성됨. |

# Part III: 프로젝트 진행중 발생한 문제점 및 해결방안

* 문제점 1: 코어 덤프로 인해 제대로 출력이 되지 않는 문제가 발생하였음
  + - 배열 길이를 초과해서 계산하려는 부분을 개선하여 문제를 해결했다.
* 문제점 2: 한글이 제대로 출력되지 않음.
  + - locale.h 헤더 파일을 가져오고 setlocale() 을 이용하여 한글을 표기할 수 있게 하였다.

# Part IV: 개선방향

[끝]