

## 실습 2. 프로그래밍 환경, 디버깅 및 기본 함수

### 1. 실습의 목표

본 실습에서는 리눅스에서 C, C++, Rust 프로그래밍 개발 및 디버깅 환경 및 C 언어 기본 라이브러리 함수에 대한 기본 개념을 확립하고 프로그래밍 및 설계 능력을 갖추는 것이 주요 목표이다.

### 2. 실습에 필요한 기초 지식

본 실습에서는 gcc, make, svn, git 및 C 언어 프로그래밍에서 디버깅에 대한 이해와 gdb 및 vscode 디버거 사용 환경에 대한 지식을 필요로 한다. 그리고, C, C++, Rust 언어 라이브러리 표준 및 기본 라이브러리 함수들에 대한 이해와 지식을 필요로 한다.

### 3. 실습에 필요한 장비

#### 하드웨어

1	PC	Host Computer
---	----	---------------

#### 소프트웨어

1	리눅스	사용 환경
1	gcc	GNU 컴파일러 패키지
1	make	프로그래밍 도구
1	svn	버전 관리 시스템
1	git	버전 관리 시스템
1	gdb	GNU 디버거
1	vscode	GUI 통합개발환경

### 4. 실습에 필요한 준비 사항

1. 교재에서 C 프로그래밍 환경에 대한 내용을 읽고 이해한다..
2. GNU autotools (autoconf, automake, libtool) 과 CMake 에 대하여 조사하고 사용 방법을 보고서에 정리하여 본다.
3. RCS (Revision Control System) 에 대하여 조사하고 사용 방법을 보고서에 정

리하여 본다.

4. 교재 등을 읽고 디버깅 및 오류 처리에 대해 내용을 이해한다.
5. Valgrind 에 대해 조사하고 요약하여 보고서에 정리하여 본다.
6. 교재에서 POSIX 표준 및 기본 라이브러리 함수에 대한 내용을 읽고 이해한다..
7. GNU C 라이브러리(glibc) 에 대하여 조사하고 요약하여 보고서에 정리하여 본다.
8. C++ STL 에 대해서 조사하고 요약하여 보고서에 정리하여 본다.
9. Rust 에 대해서 조사하고 요약하여 보고서에 정리하여 본다.

## 5. 실습 순서

1. 사칙연산 (덧셈 add, 뺄셈 subtract, 곱셈 multiply, 나눗셈 divide) 에 대한 함수와 이 함수들을 이용하는 예제 프로그램을 각각 작성하고, 정적 라이브러리, 공유 라이브러리, 동적 라이브러리 방식을 각각 이용하여 예제 프로그램을 실행시킨 결과를 보이시오.
2. 위 문제에 대하여, C 소스 파일들을 목적 파일로 만들고, 또한 이 파일들을 이용하여 라이브러리 또는 실행 파일로 만들어 주는 Makefile을 작성하여 실행시켜 보시오. 이 때, 라이브러리 관련 파일들은 서브디렉토리에 두도록 한다.
3. GIT 의 사용법을 정리하고, Github 사이트에 자신의 계정을 만들고 위아래의 모든 과제를 lab2 프로젝트에 올리시오.
4. GDB 와 Vscod의 사용법을 정리하고, 위 2번 실습 예제에 대하여 GDB 및 Vscod 테스트 결과를 나타내시오.
5. perror() 함수 구현 : 라이브러리 함수인 perror() 함수와 동일하게 동작하는 my\_perror() 함수를 구현하고, 이를 이용하여 예제 13에서 perror() 함수를 대체하여 프로그램을 실행해 보자.
6. assert() 함수 구현 : 라이브러리 함수인 assert() 함수와 동일하게 동작하는 my\_assert() 함수를 구현하고, 이를 이용하여 5장 예제 14에서 assert() 함수를 대체하여 프로그램을 실행해 보자.
7. 위 1-2번 실습문제의 프로그램에 대하여 gprof 프로파일링과 Valgrind 메모리 누수 디버깅을 적용하여 보시오.
8. 두 행렬의 크기를 입력받아서 필요한 만큼의 메모리를 동적으로 할당하여 행렬의 내용을 저장하고 두 행렬을 더하는 프로그램을 작성하시오.
9. 위 8번 문제를 C++ STL 의 vector 컨테이너와 Rust 의 Vec<T> 자료형을 활용하여 구현하시오.
10. 학생 수를 입력받은 다음, 학생의 이름, 중간, 기말 점수를 기록하는 구조체를 학생 수만큼 동적으로 할당받아서, 이를 트리 구조체로 활용하여 트리 탐색을

할 수 있는 프로그램을 작성하시오.

11. 위 10번 문제를 C++ STL 의 map 컨테이너와 Rust 의 BTreeMap<T, V>을 활용하여 구현해 본다.

## 6. 검토

실습의 요점, 결과, 검토 사항에 대해 기술하라.