2주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1. 목적**

UNIX 상에서 제공하는 C/C++관련 도구를 미리 사용해 봄으로써, 수업시간에 실습이 원활히 진행될 수 있도록 한다.

**2. 예비 학습**

본문을 읽고 c/c++ 프로그램의 컴파일 과정에 대하여 요약하라. 각 단계별로 하는 일들 과 관련된 도구들 또한 명시하라.

컴파일 과정은 아래와 같은 단계를 거친다.

Program 🡪 **<1. Preprocessor>** 🡪 **<2. Compiler>** 🡪 **<3. Assembler>** 🡪 **<4. Linker>** 🡪 Executable file

**1. Preprocessor 과정 (.c, .cc, .ccp 🡪 .i)**

-Compiler가 Compile을 잘할 수 있도록 프로그램 소스를 재구성하는 역할

**(1) 주석 제거**

-소스내에 있는 모든 주석은 제거한다.

**(2) 헤더파일 처리**

-(#include) : 표준 헤더파일 및 사용자가 작성한 헤더 파일의 내용을 가져와 삽입한다.

**(3) 매크로 치환**

-(#define) : 소스내에 치환된 식별자를 사용한 부분은 다시 치환 전의 특정 단어 및 함수로 돌려놓는다.

**2. Compiler 과정 (.i 🡪 .s)**

-언어의 문법을 검사하고, Static 영역의 메모리를 할당하면서 소스를 어셈블리어로 변환한다.

-구조

**(1) 프론트 엔드(Front-end)**

-소스의 어휘,구문,의미가 제대로 사용되었는지 검사하고, 미들 엔드로 넘겨주기 위해 트리를 구성한다.

**(2) 미들 엔드(Middle-end)**

-프론트 엔드에게 받은 트리를 이용해 CPU 상관없이 할 수 있는 최적화(아키텍처 비종속적인 최적화)를

수행한 뒤 백 엔드에서 사용하는 RTL(Resister Transfer Language)를 생성한다.

**(3) 백 엔드(Back-end)**

-미들 엔드에게 받은 RTL을 미들 엔드와 달리 아키텍처 종속적인 최적화를 하여

최종적으로 어셈블리어 코드를 생성한다.

**\*Compile 후 gcc가 삭제하지만, -s 옵션을 통해 남길 수 있다.**

**3. Assembler 과정 (.s 🡪 .o)**

-어셈블리어 코드를 컴퓨터가 이해할 수 있도록 Assembler를 통해 기계어로 변환하여

오브젝트 코드를 생성한다.

-Object File Format

**(1) Object File Header :** 오브젝트 파일의 기초적인 정보를 가짐

**(2) Text Section :** 기계어로 변환된 코드가 있는 부분

**(3) Data Section :** Data가 있는 부분

**(4) Symbol Table Section :** 소스코드에서 Symbol의 이름과 주소가 Define되어 있는 부분

**(5) Relocation Information Section :** Symbol의 위치가 정해지면 바꿔야 할 정보가 있는 부분

**(6) Debugging Information Section :** Debugging이 필요한 정보가 있는 부분

**\*Compile 후 gcc가 삭제하지만, -c 옵션을 통해 남길 수 있다.**

**4. Linker 과정 (.o, .a 🡪 .out)**

-여러 오브젝트 파일을 하나로 합치거나 라이브러리를 합쳐 실행 파일을 만드는 역할

-Link하는 방법

**(1) Static Linking :** 모든 라이브러리 모듈을 복사해서 사용

**(2) Dynamic Linking :** 라이브러리 모듈의 주소를 가지고 있다가 필요한 모듈만 가져와서 사용

-역할

**(1) Symbol Resolution :** Symbol Reference를 어떤 Symbol Definition에 연결할지 결정

**(2) Relocation :** Data의 주소나 Memory Reference 주소를 알맞게 배치