CSE3013 (컴퓨터공학 설계 및 실험 I)
UNIX-2 예비 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 목적

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

2 예비 학습

C/C++ 소스 파일의 빌드 과정은 **프리프로세싱**preprocessing, **컴파일링**compiling, **어셈블링**assembling, **링킹**linking으로 구성된다.

2.1 프리프로세싱

프리프로세싱은 선행 처리기라고도 불리는 **프리프로세서**preprocessor에 의해 이루어진다. 이름에서 보이듯이, 코드를 컴파일러가 처리하기 전에 특정 변수를 미리 정의된 문자열로 치환하는 등의 작업을 수행한다.

C/C++에서는 매크로macro와 매크로 확장macro expansion을 정의할 수 있다. 사용예는 다음과 같다.

```c
#include <stdio.h>
#define PI 3.141592653
#define area(x) (PI * (x) * (x))

int main() {
    printf("%f\n", area(10.0f));
    return 0;
}
```

이 코드는 프리프로세서에 의해 컴파일러가 처리하기 전에 다음과 같이 바뀐다.

```c
#include <stdio.h>

int main() {
    printf("%f\n", (3.141592653 * (10.0f) * (10.0f)));
    return 0;
}
```

2.2 컴파일링

컴파일은 **컴파일러**compiler에 의해 이루어진다. C/C++ 소스 파일은 컴파일러를 통해 어셈블리 소스 파일이 된다. '컴파일'은 이 과정이지만, 일반적으로 어셈블링과 묶여서, 혹은 소스가 실행 파일이 되는 전 과정을 묶어서 '컴파일'이라고 부르기도 한다. 대표적인 컴파일러들로는 gcc GNU Compiler Collection, LLVM, 그리고 MS Visual C++ compiler가 있는데, 이들은 프리프로세서의 역할도 한다.

컴파일 과정은 전단부front-end, 중단부middle-end, 후단부back-end로 구성된다. 이 전 과정이 실행되는 시간을 **컴파일 타임**compile time이라고 한다.

전단부에서는 **렉서**lexer(어휘 해석기)가 소스코드를 토큰token으로 나누면서 문법적 오류를 검출하고, 이 토큰들로 **파서**parser(구문 해석기)가 파스 트리parse tree를 구성하여 기능적 오류를 검출한다. 마지막으로 언어에 비종속적인 GIMPLE 트리를 만들어 중단부에 제공한다.

중단부에서는 GIMPLE 트리를 SSAStatic Single Assignment 형식의 자료로 변환한 후, 환경에 종속적이지 않은 최적화를 진행한다. 여기서 환경이랑 OS 혹은 CPU 아키텍쳐를 말한다. 최적화가 완료되면 후단부에 RTLRegister Transfer Language 형식으로 제공한다.

후단부에서는 RTL 최적화 이후 환경에 종속적인 최적화를 수행한다. 최적화가 완료되면 코드 생성기가 어셈블리 코드를 생성하고, 최종적으로 어셈블러에 코드를 넘겨주게 된다.

gcc의 경우 `-O0`, `-O1`, `-O2`, `-O3`, `-Os`, `-Og`, `-Ofast` 등의 플래그로 최적화 단계를 지정할 수 있다.

2.3 어셈블링

컴파일된 코드는 **어셈블러**assembler가 한 번 더 처리한다. 이 때 어셈블러가 생성하는 파일을 목적 코드object code라고 한다. 이 목적 파일은 명령어와 자료가 저장된 ELF 바이너리 포맷이다.

어셈블러는 주로 어셈블리 니모닉mnemonics을 op-code로 해석하는 작업을 하며, 컴파일러보다 구조가 훨씬 간단하다. 어셈블러가 실행되는 시간을 **어셈블리 타임**assembly time이라고 한다.

2.4 링킹

프리프로세싱, 컴파일링, 어셈블링을 거친 코드는 마지막으로 **링커**linker에 의해 라이브러리와 링킹 된다. 링커는 목적 파일들과 프로그래머가 프로그램에서 명시한 표준 C 라이브러리(`stdio.h` 따위의), 사용자 라이브러리 등을 링크link한다. 만들어진 목적 파일들은 링킹 과정을 거치면서 최종적으로 OS가 런타임에서 실행할 수 있는 실행 파일executable file이 된다.