

14장

실행 가능한 프로그램 작성

14.10 탐구문제

14.20 gcc를 사용할 수 있다면 RTL 중간 코드를 덤프하게 하는 다양한 컴파일 시점 플래그를 사용해 gcc를 실행해보자. gcc 컴파일러 버전 3.3은 이러한 플래그를 30개나 지원한다. 모든 플래그가 `-dX`의 형태를 가지는데 여기서 `X`는 `ri h j X e s F G L t b f c C N S l g o w z n E R k B M d k` 중 하나다. 여기서 보인 순서대로 플래그들은 점차적으로 더 진행된 RTL을 덤프한다. 주변의 유닉스 전문가에게 도움을 청해 `gcc.info` 파일을 찾고 접근해보자. 이 파일은 RTL과 다양한 컴파일러 단계를 기록한다.

14.21 가장 즐겨 쓰는 운영체제에서 링킹이 어떻게 동작하는지 알아보자. 코드가 동적으로 링킹될 수 있는가? (특정 권한이 없는) 사용자가 공유 라이브러리를 생성할 수 있는가? 로더는 프로그램에 링킹될 라이브러리를 어떻게 찾는가? 컴파일러가 위치 독립적 코드를 생성하게 지시할 수 있다면 이 코드는 위치 독립적이지 않은 코드와 비교해 (크기와 실행시간 효율성 측면에서) 어떻게 다른가?

14.22 본래 충분하지 않은 가장 주소 공간을 가진 기계상에서 프로그램을 실행하기 위해 개발된 포인터스위즐링(`pointerswizzling`)[Wil92a]에 대해 공부해보자. 이와 동적 링킹의 관련성을 설명하라.

14.23 자바 바이트 코드와 C#과 기타 .NET 언어가 사용하는 공통 중간 언어(CIL, Common Intermediate Language)를 비교하고 대조하라.