20160074 화학공학과 고진민

[컴퓨터 SW시스템 개론](http://lms.postech.ac.kr/Main.do?cmd=viewCourseMain&mainDTO.courseId=20180920140072901661213&gubun=class_learner) HW2

1. IEA는 명령어 집합 구조라고도 하며 마이크로프로세서가 인식하는 어셈블리어 명령어를 의미한다. 가장 low-level한 프로그래밍 interface로서, 프로세서가 실행 가능한 모든 명령어들을 포함한다. 이들의 종류에는 자료형, 명령어, 레지스터, 어드레싱 모드, 메모리 구조, 인터럽트, 예외 처리, 외부 입출력 등이 있으며 이는 프로그래밍 관련 컴퓨터 아키텍처의 일부이다.
2. RISC는 Reduced Instruction Set Computer의 약자로, 축소 명령어 집합 컴퓨터를 의미한다. 이 방법은 CPU의 명령어의 수를 줄여 하드웨어의 구조를 보다 간단하게 만드는 방법이다. CISC는 Complex Instruction Set Computer의 약자로, 복잡 명령어 집합 컴퓨터를 의미한다. 전통적인 CISC CPU에는 명령어와 주소 모드가 매우 많으나 이 중 쓰이는 것들은 별로 없다. 이 점에 착안하여 보다 적은 수의 명령어만으로 IEA를 구성한 것이 RISC이다.
3. Conditional move, 명령어로는 cmov가 좋은 이유는 효율적이기 때문이다. Conditional move는 branch misprediction을 막기에 좋으며, 이를 통해 낭비되는 사이클을 막을 수 있다.
4. 만일 xp가 NULL pointer라면, movq에서 일어나는 역참조(dereference) 과정에서 문제가 발생하게 된다.
5. (3.59)

힌트를 보면, 128bit로 확장한 경우의 표현이 x=264∙xh+xl, y=264∙yh+yl 이다. 이 둘의 곱은 다음과 같이 표현된다. x∙y=2128∙xh∙yh+264(xh∙yl+xl∙yh)+xl∙yl 여기서 이 값이 곱의 결과값인 p=264∙ph+pl 과 동치여야 하므로 ph= xh∙yl+xl∙yh, pl=xl∙yl 이다.

먼저 rdx에는 64bit int y가 저장되어 있고 rsi에는 64bit int x가 저장되어 있다. Line 2에서 rax에 rdx를 옮기고, 이를 line 3에서 cqto로 128bit로 sign extend한다. 즉, line 3에서 y를 128bit로 변환하는 것이고 이것이 C code에서 y를 int128\_t로 명시적 형변환을 하는 과정이다. 그러면 rdx에 y의 부호비트가 채워지고 rax에는 y가 저장되어 있다.

Line 4에서 rcx에 rsi, 즉 x를 옮기고 line 5에서 sarq 과정을 통해 arithmetic right shift를 63bit 만큼 진행한다. 즉, rcx에는 x의 부호비트가 채워진다. 다음으로 line 6에서 rcx와 rax를 곱해 rcx에 저장하고 line 7에서 rdx와 rsi를 곱해 rdx에 저장한다. 이 결과 rcx에는 y\*(x의 부호비트로 채워진 값)이 저장되고, rdx에는 x\*(y의 부호비트로 채워진 값)이 저장된다.

Line 8에서 rdx와 rcx를 더해 rcx에 저장한다. 그리고 rax에 rsi와 rax의 곱을 저장한다. 이는 x와 y의 곱이다. 이후 rdx에 rdx와 rcx의 합을 저장하고 이 값이 위에서의 ph가 된다. 그리고 rax에 저장된 값이 pl이 된다.

1. (3.62)

소스 코드 참조 요망

1. (3.63)

소스 코드 참조 요망