

13장. 프로세서 (Processors)



리눅스는 여러 프로세서에서 실행된다. 이 장은 이들을 간단히 설명한다.

13.1 X86

TBD

13.2 ARM

ARM 프로세서는 저전력 고성능의 32비트 RISC 아키텍처이다. ARM은 이동 전화, PDA(Personal Data Assistant)와 같은 임베디드(embedded) 장치에서 널리 사용되고 있다. ARM 은 31개의 32비트 레지스터를 가지고 있으며 각 모드에서 16개를 사용할 수 있다. 명령어 는 단순한 load와 store명령 (메모리에서 값을 가져오고, 계산하고, 결과를 메모리에 저장한 다) 위주로 구성되어 있다. 한가지 재미있는 특징은 모든 명령이 조건부 명령이라는 것이다. 예를 들어 어떤 레지스터의 값을 테스트한 후, 다시 같은 조건을 테스트할 때까지, 테스트 결과에 따라 원하는 대로 조건부 명령을 실행할 수 있다. 또다른 재미있는 특징은 값을 메모리에서 로드하면서 산술/쉬프트 연산을 동시에 할 수 있다는 것이다. ARM은 사용자 모드 와, 여기서 SWI(소프트웨어 인터럽트)를 통해 들어갈 수 있는 시스템 모드를 포함하여 여러 모드에서 동작한다.

ARM은 합성을 위한 핵심이며, ARM사는 직접 프로세서를 생산하지는 않는다. 대신에 ARM 파트너(인텔이나 LSI 등)가 ARM 아키텍처를 실리콘으로 구현한다. 이 방식은 다른 프로세서가 보조프로세서 인터페이스를 통해 긴밀하게 결합될 수 있도록 하며, 여러 종류의 메모리 관리 유닛(memory management unit, MMU)의 변형을 갖고 있다. 이들은 단순한 메모리 보호 정책부터 복잡한 페이지 계층구조에까지 이른다.

13.3 알파 AXP 프로세서

알파 AXP 아키텍처는 64비트 load/store RISC 아키텍처로서 속도를 염두에 두고 설계되었다¹. 모든 레지스터는 64비트로, 32개의 정수 레지스터와 32개의 실수 레지스터가 있다. 31번 정수 레지스터와 31번 실수 레지스터는 null 연산을 위해 사용된다. 이들 레지스터를 읽으면 0이 돌아오고, 이들에 값

을 쓰는 것은 아무런 효과도 없다. 모든 명령은 32비트이며 메모리 연산은 읽기 아니면 쓰기이다. 이 아키텍처는 구현이 아키텍처를 따르는 한 여러가지 구현 을 허용한다.

여기에는 메모리에 저장된 값을 직접 연산 대상으로 하는 명령은 없다. 모든 데이터 처리는 레지스터 간에 이루어진다. 따라서 메모리의 카운터 값을 증가시키고 싶으면 먼저 레지스터 에 읽어온 다음 값을 변경하고 메모리에 기록해야 한다. 명령들간의 상호작용은 한 명령이 값을 레지스터나 메모리에 쓰고, 다른 명령이 그 레지스터나 메모리에서 읽어오는 것을 통 해 이루어진다. 알파 AXP의 한가지 재미있는 특징은, 두 레지스터 값이 같은가를 테스트하 는 것과 같이 플래그를 발생시키는 명령의 결과가, 프로세서 상태 레지스터에 저장되는 것 이 아니라 제3의 레지스터에 저장할 수 있는 명령이 있다는 점이다. 처음 보기에는 이상하 지만 상태 레지스터에 대한 의존을 제거함으로써 각 사이클에 여러 명령을 실행할 수 있는 프로세서를 만들기가 더욱 쉬워진다. 서로 관계없는 레지스터를 사용하는 명령은 하나의 상 태 레지스터가 있을 때처럼 실행을 위해 서로를 기다릴 필요가 없다. 메모리에 대한 직접 연산이 없는 것과, 레지스터의 수가 많은 것도 여러 명령을 동시에 실행하는데 도움이 된다.

알파 AXP 아키텍처는 PALcode(특권 아키텍처 라이브러리 코드)라고 불리는 서브루틴들을 사용한다. PALcode는 운영체제, 알파 AXP 아키텍처를 갖는 CPU 구현, 시스템 하드웨어에 따라 다르다. 이들 서브루틴은 컨텍스트 스위칭(context switching), 인터럽트, 예외(exception), 메모리 관리 등의 운영체제 프리미티브를 제공한다. 이들 서브루틴은 하드웨어나 CALL_PAL 명령에 의해 호출될 수 있다. PALcode는 내부 프로세서 레지스터와 같은 저수준 하드웨어 기능에 대한 직접 접근을 제공하기 위해, 구현에 따른 약간의 확장을 포함한 표준 알파 AXP 어셈블러로 작성된다. PALcode는 PALmode에서 실행된다. 이 모드는 몇가지 시스템 이벤트의 발생을 중지시키고 PALcode가 실제 시스템 하드웨어에 대한 제어를 완료하도록 하는 특권 모드이다.

번역 : 심마로
정리 : 이호

역주 1) 높은 클럭을 제공할 수 있는 설계로 인해 21264이전에는 다른 RISC CPU에 비해 클럭당 성능 면에서는 좋지 못했다. (심마로)