1장. 하드웨어의 기초

운영체제는 그의 기반이 되는 하드웨어 시스템과 밀접한 관계를 가지고 동작해야 한다. 운 영체제는 하드웨어만이 제공할 수 있는 특정 서비스들을 필요로 한다. 리눅스 운영체제를 완전히 이해하려면 이의 기반이 되는 하드웨어의 기본 사항들을 이해하고 있어야 한다. 이 장에서는 하드웨어 - 요즘의 PC - 에 대해 간단히 소개하도록 하겠다.

"Popular Electronics" 잡지의 1975년 1월호 표지에 알테어(Altair) 8080의 삽화가 등장했을 때부 터 혁명은 시작되었다. 스타트렉 초기 에피소드에 등장하는 목적지의 명칭을 따서 이름지어 진 알테어 8080은[1], 취미로 전자 공작을 즐기는 열성파들이 겨우 397 달러만 들이면 조립할 수 있는 것이었다. 인텔 8080 프로세서와 256 바이트의 메모리에 화면과 키보드도 없어 요 즘 기준으로는 보면 보잘것없는 것이다. 이것을 개발한 에드 로버트(Ed Roberts)는 자신의 새 발명품에 "개인용 컴퓨터(personal computer, PC)"라는 이름을 붙였는데, 이제 이 PC라는 용어는 혼자서 들 수 있는 크기의 대부분의 컴퓨터를 가리키게 되었다. 이 정의에 따르면 매우 강력한 성능을 발휘하는 알파 AXP 시스템 역시 PC라고 할 수 있다.

열렬한 해커들은 알테어의 잠재력을 알아보았고, 이를 위한 소프트웨어를 작성하고, 하드웨 어를 제작하기 시작했다. 그것은 이들 초기 선구자들에게 있어 자유 - 엘리트 성직자에 의 해 실행되고 보호되는 거대한 일괄처리 메인프레임 시스템으로부터의 자유 - 를 의미했다. 자기집 식탁 위에 놓을 수 있는 컴퓨터라는 이 새로운 현상에 고무된 대학 중퇴자들은 순식 간에 큰 돈을 벌게 되었다. 조금씩 다른 수많은 하드웨어가 등장했고, 소프트웨어 해커들은 이 새로운 기계용으로 소프트웨어를 만들 수 있어서 행복했다. 역설적이게도 요즘의 PC 형 태의 기반을 만든 것은 1981 년 IBM PC를 발표하고 1982년 초 이를 고객들에게 판매하기 시작한 IBM이었다. 인텔 8088 프로세서에 64K 메모리 (256K까지 확장가능했다), 두 개의 플 로피 디스크 드라이브와 가로 80글자, 세로 25줄의 문자를 표시할 수 있는 CGA (Color Graphic Adapter)[2] 카드를 장착한 이 컴퓨터는 요즘 기준으로 본다면 별로 강력하진 않지만 매우 잘 팔렸다. 이를 이어 IBM은 1983년, 당시엔 사치품으로 여겨진 10M 바이트의 용량의 하드 디스크가 달린 IBM-XT를 내놓았다. 오래지않아 컴팩(Compaq)을 포함한 많은 회사들이 IBM PC를 모방한 컴퓨터들을 생산하게 시작했고, 이 PC의 구조는 사실상 표준이 되었다. 이 실질적인 표준은 수많은 하드웨어 업체들이 성장단계의 시장을 놓고 경쟁하게 만들었고, 이로 인해 낮아진 가격에 고객들은 좋아했다. 이 초창기 PC가 가진 시스템 구조적 특징 중 많은 것들이 지금의 PC에까지 그대로 이어져 왔다. 예를들어, 가장 강력한 인텔 펜티엄 CPU를 채용한 시스템조차도, 처음 시작할 때 인텔 8086의 어드레싱 모드[3]에서 시작한다. 리 누스 토발즈가 나중에 리눅스라고 불리게 된 프로그램을 짜기 시작했을 때, 그는 당시 가장 널리 보급되어 있었고, 가격도 적당한 하드웨어였던 인텔 80386 PC를 선택했다.

PC의 외관을 보면, 가장 분명하게 구분할 수 있는 것은 시스템 박스와 키보드, 마우스, 그리 고 모니터이다. 시스템 박스의 앞면에는 몇 개의 버튼과, 숫자를 보여주는 작은 디스플레이[4], 그리고 플로피 드

라이브가 있다. 요즘에 나온 대부분의 시스템에는 CD ROM이 달려있고, 데 이터 보호를 필요로 하는 경우 백업용 테입 드라이브도 있을 것이다. 이들 장치들을 총괄하 여 주변장치라고 한다.

CPU가 시스템 전체를 통제하긴 하지만, CPU만이 시스템에서 지능을 가진 유일한 장치는 아 니다. IDE 컨트롤러 같은 주변장치 컨트롤러 모두 어느정도 수준의 지능을 가지고 있다. PC 내부에는 (그림 1.1) CPU(또는 마이크로프로세서라고 한다)와 메모리, 여러개의 ISA나 PCI 주변장치 컨트롤러를 꽂을 수 있는 슬롯을 갖춘 마더보드가 있다. IDE 디스크 컨트롤러같은 몇몇 컨트롤러는 시스템 보드상에 있기도 하다.

1.1 CPU

CPU(CPU보다는 마이크로프로세서란 이름이 더 적당하다)는 모든 컴퓨터 시스템의 핵심이 다. 마이크로프로세서는 메모리에서 명령을 읽고 이를 수행함으로써, 계산을 하고 논리 연산 을 수행하고, 데이터 흐름을 관리한다. 컴퓨터가 등장한 초창기에는 마이크로프로세서의 이 런 기능들이 각각 별도의 장치로 (실제로 큰 덩치의 장치로) 되어 있었다. 이 때는 중앙처리 장치(Central Processing Unit, CPU)라는 말이 적합했다. 지금의 마이크로프로세서는 이들 기능 요소들을 결합해 매우 작은 실리콘 조각 하나에 집적회로로 가지고 있다. 이 책에서는 CPU, 마이크로프로세서(microprocessor), 프로세서(processor)라는 용어를 모두 같은 의미로 사용한 다.

마이크로프로세서는 0과 1의 결합인 이진 데이터로 동작한다. 이 0과 1은 꺼진 상태와 켜진 상태를 갖는 전기스위치와 같은 것이다. 십진수로 42가 10짜리 4개와 1짜리 2개를 의미하는 것처럼, 이진수는 각각의 이진 숫자가 2의 몇제곱승을 나타내는 2진 숫자의 연속이다. 여기 서 몇제곱승이란 같은 숫자를 여러번 곱하는 횟수를 말한다. 10의 1 제곱승(10^1)은 10이고, 10 의 2제곱승(10^2)은 10x10, 10^3은 10x10x10 등등이다. 이진수 0001은 십진수로 1, 이진수 0010 은 십진수 2, 이진수 0011은 십진수 3, 이진수 0100은 십진수 4에 해당한다. 따라서 십진수 42는 이진수로 101010, 즉 2 + 8 + 32 또는 2^1 + 2^3 + 2^5이다. 컴퓨터 프로그램에서는 일반적으 로 숫자를 나타내는데 이진수를 쓰기 보다는 다른 진법인 십육진수를 사용한다. 십육진법에 서는 각 숫자가 16의 몇제곱승을 나타낸다. 숫자는 0부터 9까지만 있으므로 10부터 15까지 는 문자 A, B, C, D, E, F로 표시한다. 예를들어 십육진수 E는 십진수로 14이고, 십육진수 2A 는 숫자 42(16짜리 2개 + 10)이 된다. C 프로그래밍 언어에서는 십육진수 앞에 "0x"를 붙여서 구별한다. 즉 십육진수 2A는 0x2A라고 쓴다. 이 책에서는 이 표기법을 사용한다.

마이크로프로세서는 덧셋, 곱셈, 나눗셈 같은 숫자 연산과 "X가 Y보다 큰가?"같은 논리 연 산을 수행할 수 있다.

프로세서의 명령 수행은 외부 클럭에 의해 제어된다. 이 클럭을 시스템 클럭이라고 하며, 정 기적으로 클럭 펄스를 만들어 프로세서로 보내고, 각 클럭 펄스마다 프로세서는 주어진 일 을 하게 된다. 예를 들어, 어떤 프로세서는 각 클럭 펄스마다 명령어를 하나씩 처리한다. 프 로세서의 속도는 초당 시스템 클럭의 횟수로 나타내는데, 예를 들어 100MHz 프로세서는 초 당 1억번의 클럭 틱을 받는다. 그러나 프로세서마다 한번의 클럭 틱 동안 수행하는 일의 양 이 다르기 때문에, CPU의 성능을 클럭 속도로 비교하는 것은 잘못된 것이다. 하지만 모든 점들이 똑같다면, 클럭 속도가 빠른 것이 더 강력한 프로세서이다. 프로세서가 수행하는 명 령은 매우 단순한 것이다. 예를 들면 "메모리 X 위치에 있는 내용을 레지스터 Y로 읽어들 여라" 같은 것이다. 레지스터(register)는 데이터를 저장하고 연산을 하는데 사용하는 마이크 로프로세서 내부에 있는 기억장소이다. 어떤 명령은 프로세서가 하던 일을 중단하고 메모리 의 다른 위치에 있는 또 다른 명령어로 건너뛰게 하기도 한다. 이런 자그만 명령 단위는 프 로

세서가 1초에 수백만에서 심지어 수십억개의 명령어를 실행할 수 있게 하여, 지금의 프로 세서가 거의 무한한 능력을 가질 수 있게 한다.

명령어를 수행하려면 먼저 명령어를 메모리에서 가져와야 한다. 어떤 명령어는 메모리에 있 는 데이터를 참조하기도 하는데, 이 경우 메모리에서 데이터를 가져와야 하며, 데이터를 쓰 려고 하는 경우 메모리에 데이터를 저장하게 된다.

프로세서에 있는 레지스터의 크기와 갯수, 종류는 프로세서 종류마다 다르다. 인텔 486 프로 세서는 알파 AXP 프로세서와 다른 레지스터 세트를 가진다. 우선 인텔의 레지스터는 32비 트 크기지만 알파 AXP의 레지스터는 64비트이다. 그렇지만 대체로 어떤 프로세서이든 여러 개의 일반 목적 레지스터와 이보다 적은 갯수의 특수 목적 레지스터를 갖는다. 대부분의 프 로세서는 다음과 같은 특수 목적의 전용 레지스터를 가지고 있다.

- **프로그램 카운터 (Program Counter, PC)** 이 레지스터는 다음에 실행할 명령어의 주소를 가 지고 있다. 이 값은 명령어를 가져올 때마다 자동으로 증가한다.
- **스택 포인터 (Stack Pointer, SP)** 프로세서는 데이터를 임시로 저장할 수 있는 대규모의 외부 RAM에 접근해야 한다. 스택은 외부 메모리에 임시로 데이터를 저장하고 다시 읽어들일 수 있는 손쉬운 방법 중 하나이다. 대개 프로세서들은 스택에 데이터를 넣고 (push), 나 중에 이를 다시 가져오는 (pop) 특별한 명령어들을 가지고 있다. 스택은 "마지막에 들어 온 것이 맨 먼저 나가는 (last in first out, LIFO)" 방식으로 동작한다. 다르게 말하면, 스택 에 두개의 값 x와 y를 집어넣고, 값을 빼내면 먼저 y값을 먼저 얻게 되는 것이다.

  어떤 프로세서에서는 스택이 위로 자라지만, 다른 프로세서는 메모리가 시작하는 쪽인 아래쪽으로 스택이 자란다. ARM같은 프로세서는 두가지 방식 모두를 지원한다.
- **프로세서 상태 (Processor Status, PS)** 어떤 명령어들은 실행하면 결과가 나오는 것이 있다. 예를 들어 "레지스터 X의 값이 Y의 값보다 큰가?"라는 명령을 수행하면 예 또는 아니오 의 결과가 나온다. PS 레지스터는 이런 값과 함께, 프로세서의 현재 상태를 나타내는 다 른 정보들을 가지고 있다. 이런 예로, 대부분의 프로세서는 커널 모드(또는 관리자 모드) 와 사용자 모드라는 두가지 이상의 동작모드를 가지고 있는데, PS 레지스터는 현재 어떤 모드에 있는지 나타내는 있는 정보를 가지고 있다.

1.2 메모리(Memory)

모든 시스템에는 메모리 분류 체계가 있으며, 다른 크기와 속도를 갖는 메모리들이 이 체계 의 서로 다른 지점에 위치한다. 우선 가장 빠른 메모리는 캐시 메모리로, 말 그대로 메인 메 모리의 내용을 임시로 보관하는, 즉 캐시하는데 사용하는 메모리이다. 이런 메모리는 속도는 매우 빠르지만 값이 비싸기 때문에, 대부분의 프로세서는 칩 안에 소량의 캐시 메모리를, 그 리고 보드상에 추가로 캐시 메모리를 가지고 있다. 어떤 프로세서는 하나의 캐시에서 명령 어와 데이터를 같이 갖지만, 명령어와 데이터 용으로 두 개의 캐시를 갖는 것도 있다. 알파 AXP 프로세서는 두개의 내장 메모리 캐시를 가지고 있는데, 하나는 데이터용이고(D-캐시), 다른 하나는 명령어용이다(I-캐시). 외장 캐시(B-캐시)는 이 두 가지를 함께 가진다. 마지막으 로 외장 캐시 메모리에 비해 매우 느린 메인 메모리가 있다. CPU 칩상에 있는 캐시와 비교 하면 메인 메모리는 정말 밥통같은 것이다

캐시와 메인 메모리는 같은 값을 유지하고 있어야 한다 (일치성). 다르게 말하면, 메인 메모 리에 있는 어떤 데이터가 캐시의 하나 이상의 위치에 저장되어 있을 때, 시스템은 캐시에 있는 값과 메모리에 있는 값이 일치하도록 해주어야 한다는 것이다. 캐시의 일치성은 어떤 부분은 하드웨어에 의해, 어떤 부분은 운영체제에 의해 유지된다. 이런 것은 소기의 목적을 달성하기 위해 하드웨어와 소프트웨어가 밀접하게 협동해야 하는, 시스템의 다른 주요 작업 들에 있어서도 마찬가지다.

1.3 버스(Bus)

시스템 보드상의 개개 구성요소들은 여러개의 버스라는 연결시스템으로 상호 연결되어 있다. 시스템 버스는 세가지 논리적인 기능 요소로 나누어지는데, 하나는 주소 버스(address bus), 다른 하나는 데이터 버스(data bus), 나머지 하나는 제어 버스(control bus)이다. 주소 버스는 데이터를 전송할 메모리의 위치(주소)를 지정한다. 데이터 버스는 전송되는 데이터를 가지고 있으며, 양방향으로 전송 가능하다. 즉 CPU로 읽어 들이거나 CPU에서 쓰는 것이 가능하다. 제어 버스는 시스템 전체에 타이밍 신호와 제어 신호를 전달하는 여러 선들을 가지고 있다. 여러 방식의 버스가 있지만, ISA나 PCI 버스가 주변장치를 시스템에 연결하는 대중적인 방 법으로 사용되고 있다.

1.4 컨트롤러와 주변장치

주변장치는 시스템 보드 상이나 또는 보드에 꽂힌 카드에 있는 컨트롤러 칩에 의해 제어되 는, 그래픽 카드나 디스크같이 실제로 존재하는 장치를 말한다. IDE 디스크는 IDE 컨트롤러 칩에 의해, SCSI 디스크는 SCSI디스크 컨트롤러 칩에 의해 제어된다. 이들 컨트롤러는 여러 종류의 버스를 통해, CPU와 다른 컨트롤러들과 서로 연결되어 있다. 요즘 나오는 시스템의 대부분은 이들 주요 시스템 구성요소들을 연결하기 위해 PCI와 ISA 버스를 사용한다. 컨트 롤러는 CPU와 비슷한 하나의 프로세서이고, CPU 입장에서는 똑똑한 도우미이다. CPU는 시 스템 전체를 제어하는 것이다.

모든 컨트롤러는 서로 다르지만, 자신을 제어하기 위한 레지스터를 가지고 있다는 점은 비 슷하다. CPU에서 실행되는 소프트웨어는 이들 제어용 레지스터를 읽고 쓸 수 있어야 한다. 어떤 레지스터는 에러를 나타내는 상태를 가지고 있기도 하고, 또다른 레지스터는 컨트롤러 의 모드를 바꾸는 것 같은 제어 용도로 사용되기도 한다. CPU는 버스상에 있는 컨트롤러 각 각에 개별적으로 주소지정을 할 수 있다. 이리하여 소프트웨어 디바이스 드라이버가 컨트롤 러를 제어하기 위해 레지스터를 쓸 수 있게 된다. IDE 리본이 좋은 예로, 이는 버스상에 있 는 드라이브를 따로따로 접근할 수 있도록 해준다. 다른 좋은 예로는 각 디바이스(그래픽카 드같은)들을 서로 독립적으로 접근할 수 있는 PCI 버스가 있다.

1.5 주소공간(Address Space)

CPU와 메인 메모리를 연결하는 시스템 버스는, CPU와 다른 하드웨어 주변장치를 연결하는 버스와는 분리되어 있다. 하드웨어 주변장치가 존재하고 있는 메모리 공간을 총괄하여 I/O 공간이라고 한다. I/O 공간은 더 쪼갤 수 있지만, 당분간 이에 대해 생각하지 않도록 하자. CPU는 시스템 공간 메모리와 I/O 공간 메모리에 모두 접근 가능하지만, 컨트롤러는 단지 시스템 메모리에 간접적으로 접근할 수 있을 뿐이며, 이것도 CPU의 도움을 받아야만 한다. 장치의 입장에서 보면, 가령 플로피 디스크 컨트롤러

라고 한다면, 자신의 제어 레지스터가 있는 주소공간(ISA)만 보일 뿐, 시스템 메모리는 보이지 않을 것이다. 일반적으로 CPU는 메 모리 공간과 I/O 공간을 접근하는데 다른 명령어를 사용한다. 예를 들어, "I/O 공간 0x3f0 주 소에서 한 바이트를 읽어 레지스터 X에 저장하라"같은 명령이 있는 것이다. 이는 CPU가 I/O 공간에 있는 주변장치의 레지스터를 읽고 씀으로써, 하드웨어 주변장치를 제어하는 방 법을 그대로 보여준다. 일반적으로 쓰이는 주변장치들(IDE 컨트롤러, 직렬포트, 플로피 디스 크 컨트롤러 등)의 레지스터가 있는 I/O 공간은 PC 구조가 개발된 후 오랫동안 관례에 의해 고정되어 있다. I/O 공간의 주소 0x3f0은 직렬포트 COM1 제어 레지스터 중 하나의 주소이 다.

가끔은 컨트롤러가 많은 양의 데이터를 시스템의 메모리에서 읽어 들이거나 메모리로 써 넣 어야 할 경우가 있다. 사용자의 데이터를 하드디스크에 기록하는 경우가 이런 좋은 예이다. 이 때는, DMA (Direct Memory Access, 직접 메모리 접근) 컨트롤러를 사용하여 하드웨어 주변 장치가 바로 시스템 메모리에 접근할 수 있게 한다. 하지만 이것 역시 CPU의 엄격한 제어 와 감시하에 이루어진다.

1.6 타이머

모든 운영체제는 현재 시간을 알 필요가 있기 때문에, 지금 나오는 PC들은 RTC(Real Time Clock, 실시간 클럭)라는 특수한 주변장치를 가지고 있다. 이것은 정확한 시간과, 정밀한 시 간 간격을 제공하는 두가지 역할을 한다. RTC 는 자체 배터리를 가지고 있어서, PC의 전원 을 끄더라도 계속 동작한다. 이것이 PC가 항상 정확한 날짜와 시간을 알 수 있는 방법이다. 간격 타이머(interval timer)는 운영체제가 중요한 작업의 일정을 정확하게 조절할 수 있게 해 준다.

번역 : 이호, 김진석, 이대현, 이준희, 고양우, truejaws
정리 : 이호

역주 1) Altair는 독수리자리의 알파별의 이름으로 우리말로 견우성이라고 한다. (jhlee)

역주 2) IBM PC 초창기에 사용했던 컬러 그래픽 카드 (flyduck)

역주 3) 인텔 8086 CPU는 모두 1M 바이트를 나타낼 수 있는 20비트 어드레싱 모드에서 동 작하며, 인텔 80386 이후의 CPU는 (펜티엄을 포함하여) 32비트 어드레싱 모드에서 4G 바 이트까지 메모리를 사용할 수 있지만 처음 시작할 때는 8086과 마찬가지로 20비트 어드 레싱 모드에서 시작한다. (flyduck)

역주 4) 486 이전의 케이스에는 터보 모드를 위하여 클럭 속도를 보여주는 LED가 달려 있 었는데, 요즘 PC에는 터보 모드라는것이 없기 때문에 요즘에 나오는 케이스에는 달려있 지 않다. (flyduck)