|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **이미지** | **해설** | |  | Intel 8088 CPU  81년도 생산된 16비트 CPU    인텔에서 좋은성능과 확장성을 위해  78년도에 x86 아키텍처 를 만들게 되는대 그것이  8086 이다  6502하고 호환이 안된다 | |  | 8088 ,8086의 제일큰 차이점은  데이터 너비 즉 한번에 명령어를 처리할 수  있는 길이가 다르다  8088은 저가형으로 널리사용되었고 IBM PC XT  에 내장되어 사용되었다 | |  | IBM PC XT | |  | - 8088 cpu 핀구성 -  AD핀  주소핀과 데이터핀을 따로 만들면 핀이 많아지게  되고 그것은 비용이 많이 들어가기때문에  하나의 핀에서 두가지를 처리하게 만들었다  주소 와 데이터 는 타이밍에 따라 주소모드  데이터모드로 사용된다  OS의 장치드라이버가 명령어를 받아서  CPU에게 데이터 전송 ,제어 ,타이밍 등 중간에서  전달 한다 | |  | - 8086 cpu 핀구성 - | |  | 추상화 단계  1) Application  어플리케이션 ( 응용프로그램 )  2) Programming Languages  프로그래밍 언어  3) Operating Systems  OS  운영체제  3.5) Bios  바이오스  4) Instruction Set Architecture  ISA  명령어형식 ,명령어셋 ,레지스터구성 ,주소지정방식  5) Microarchitecture  CPU  기능, 구성요소, 데이터플로우, 명령어셋  6) Execution Units  ControllerUnit에서 해독한 명령어를  ExecutionUnit에 할당되어  ALU에 전달하는 역활을 한다  7) Functional Units  Core  ( ALU ,ControllerUnit ,Register ,Bus )  8) Logic Gates  논리회로  9) Transistors  트랜지스터 | |  | **운영체제 ( Operating System)**  하드웨어를 직접 제어하지 않고 운영체제의  공통서비스 중 시스템 콜 ( System Call )을 통해서 하드웨어를 제어할수 있다  **시스템 콜 ( System Call )**  인터럽트가 발생하면 시스템콜이 호출되는대 이때  CPU는 평소 유저모드 에서 커널모드로 전환 된다  **유저모드 ( User Mode )**  하드웨어에 직접접근을 제한한 모드.  **커널모드 ( Kernel Mode )**  시스템콜 에서 넘어온 명령어를  장치드라이버를 통해 CPU에 직접적으로  전달하고 처리된 결과값을 반환받아 유저모드에  반환 시키면서 유저모드로 전환된다.  모든 시스템 ,하드웨어에 접근권한을 갖는다  **장치드라이버**  커널모드에서 커널이 CPU에 직접 명령어를 전달할  수 있게 하는 중재 송수신 역활을 한다.  어떤 프로그램을 실행하면 운영체제 장치드라이버가 cpu에게 실행 명령어를 전달하여 그프로그램의 리셋백터를 실행하게 한다 | | ㄹ  ROM  ROM BIOS | 8088 의 DOS 부팅 순서 - 1  1)  컴퓨터를 부팅하면  CPU의 리셋벡터 0xFFFF0 주소를 실행하고  그주소는 Rom으로 점프하여  Rom이 실행되며  Rom의 리셋벡터 0xFFFFF 부터 0xFFFF0 까지 있는  명령어를 실행한다  Rom의 0xFFFFF ~ 0xFFFF0에 있는 명령어는  RomBios로 점프하는  명령어가 저장되어 있다  RomBios에는 메인보드에 설치된  하드웨어의 리셋벡터가 저장되어 있다  0xF0000 ~ 0xA0000 범위에는  메인보드 이외의 그래픽카드 나 입출력 장치의  리셋벡터가 저장되어 있다 | |  | 8088 의 DOS 부팅 순서 - 2  2)  첫번째 Rom에있는 Bios로 점프하는  첫명령어를 실행하면  메인보드에 하드웨어 장치가 제대로 꼽혀있는지  장치확인을 한다 | |  | 8088 의 DOS 부팅 순서 - 3  3)  모든하드웨어가 정상적으로 꼽혀있다면  인터럽트 벡터 ( 0x00000 ~ 0x003FF )까지  초기화 한다  - ROM BIOS데이터 -  부팅가능한 장치가있는지 확인을 한다  (HDD,플로피디스크)  만약 부팅가능한 장치가 있다면 Rom의 제어권을  운영체제 로더 ( Boot Sector )로 넘긴다  - Rom\_Bios -  읽기전용Rom칩에 Bios명령어를 저장한 칩  인터럽트 벡터테이블  주소 0000부터 256개의 어느정도까지 점프테이블을 저장한다  예로 0번이라는 이벤트가 발생하면  인터럽트 벡터의 0번쨰 주소에가서 0번째에 있는 주소로 점프한다 점프해서 가면 그곳에 0번을 처리할 수 있는 코드가 있다  운영체제는 ram에 저장된다 | | 플로피디스크 | 8088 의 DOS 부팅 순서 - 4  4)  만약 ROMBIOS데이터 영역에  MSDOS가 깔려있는 플로피디스크 장치가 있었다면  부트섹터 ( Boot Sector ) 영역의  첫번째 섹터를 실행하고 끝나면  운영체제로더 ( Operation System Loader ) 를  실행한다  - 부트섹터 ( Boot Sector ) -  운영체제가 설치된 플로피디스크의 첫번째섹터 이다  플로피디스크의 첫번째 섹터에는  부트로더 ( Bootloader )가 있다  - 부트로더 ( Bootloader ) -  플로피디스크에 저장된 운영체제를 부트로더가  Ram으로 복사한다  //수정필요  - Rom 로더 ( Loader ) -  Rom에 저장된 프로그램이나 데이터를  Ram으로 복사 한다  ROM에 저장된 내용은 일반적으로 수정이 불가능한데, ROM 로더는 이러한 ROM에 저장된 프로그램이나 데이터를 메모리로 복사해와서 실행 가능한 상태로 만들어주는 역할을 한다 | |  | 8088 의 DOS 부팅 순서 - 5  5)  운영체제로더 ( Operation System Loader ) 는  운영체제를 부팅하고 실행되기 위해 추가적으로  필요한 데이터 장치드라이버 ( IO.SYS )  ,MSDOS.SYS를 Ram에 복사한다  이러한 Ram에 복사를 하기위해서는  RomBios의 인터럽트를 사용해서 복사한다  여기까지 RomBios의 초기화 루틴은 끝이고  부트섹션은 삭제되고  운영체제시스템으로 제어권을 넘긴다 | |  | 8088 의 DOS 부팅 순서 - 6  운영체제 부팅에 필요한 데이터를 초기화한다  초기화가 끝나면 SYSINIT은 삭제된다  SYSINIT은 장치드라이버 ( IO.SYS )의  일부였고 부팅을위해  SYSINIT을 복사한 것이다  - 도스커널 ( Dos Kernel ) -  RomBios의 인터럽트벡터 기능을  도스에서 제공하는 함수들을 사용해  MS도스시스템 ( MSDOS.SYS ) 에서 사용할 수 있다  MSDOS를 제어할 수 있는 cmd를  Ram에 복사하고 cmd를 실행한다 | |  | 8088 MS-DOS의 문제점  기본적으로 리얼모드 라서  하드웨어를 직접제어 할 수 있어서  실행되고 있는 프로그램의 메모리를 건들수도  있기때문에 오류가 생길 가능성이 있다  심지어 MS-DOS커널 메모리도 변경을할 수 있어서  블루스크린이 생기던 시절 이다  8088다음 세대에 이러한 문제점을 해결하기  위해 보호모드가 등장한다. | |  | - 보호모드 ( Protected Mode ) -  유저모드와 커널모드를 분리시켰다  286 ,386 컴퓨터부터 보호모드를 사용했다  Level 0 : 운영체제 커널  Level 3 : 응용프로그램  레벨3이 안쪽레벨로 접근을 직접적으로 하지  못하도록 권한설정을 만들었다  이로인해 커널을보호 할 수 있었고  페이징을 통해 가상 메모리를 구현 했다 | |  | - 운영체제 없이 실행되는 프로그램 -  운영체제 설치USB  ,디스크만 넣으면 실행되던 게임 | |  | .DOSSEG  도스세그먼트의 약자  .8086  어떤 CPU를 사용할지 정의 8086과 8088은 같다  .NO87  부동소숫점을 계산하는 콤프레션을 사용할지 정한다  .MODEL  세그먼트 메모리를 간단하게 64비트 한블럭에 저장  TINY  .COM파일을 만들어준다  .DATA  데이터가 들어있는 세그먼트  .CODE  코드들이 들어있는 세그먼트  .STARTUP  MASM에서 메모리의 0x100h ( 256 ) 에  저장 하는 라벨 이다  시작하는 프로그램의 시작주소 이고  ( 메인함수 ) 이다  END  여기까지 어셈블해라 현재 파일의 끝을 표현  ML  MASM의 exe파일 컴파일하는 명령어 | |  | mov  대입 연산  lea  주소를 로드한다  int  a레지스터에 원한는 인터럽트 함수를 넣어놓고  int로 어떤 인터럽트 함수를 사용할지 정한다  리턴이 없기때문에 인터럽트를 통해 프로그램을  종료해야한다  ah  a레지스터의 16비트 하이8비트  al  a레지스터의 16비트 로우 8비트  ax  a레지스터의 16비트  h  상수뒤에 h는 헥스  b  상수뒤에 b는 2진수  o  상수뒤에 o는8진수 | |  | - 범용레지스터 -  AX ( Primary Accumulator )  ALU에서 주로 사용하는 레지스터  BX ( Base ,Accumulator )  간접메모리주소로 사용되는 레지스터  CX ( Counter ,Accumulator )  루프 ,ALU 용도로 주로사용 되는 레지스터  DX ( Accumulator ,Extended Accumulater )  ALU에서 주로 사용하는 레지스터  BP ( Base Pointer )  현재스택의 아래를 가르킴  SP ( Stack Pointer )  현재스택의 위치를 가르킴  SI ( Source Index )  for문의 i색인자  DI ( Destination Index )  for문의 i색인자 | |  | - 변수 및 초기화 -  DB ( Define Byte )  1바이트 공간  DW ( Define Word )  2바이트 공간  DD ( Define DoubleWord )  4바이트 공간  DQ ( Define QuadWord )  8바이트 공간  ?  초기화 되지 않은 데이터  메모리공간만 할당해놓고 프로그램 실행중 값이  할당될때 사용  .ALIGN  메모리정렬을 한다  .EVEN  메모리정렬을 짝수 2바이트씩 한다 | |  | 배열선언  문자열  형지정자 로 배열사용 | |  |  | | | |