시스템 부팅 절차(System Boot Sequence)

이번에는 Windows NT 시스템을 부팅한 후부터 콘솔상에 Windows Log On 화면이 나타날 때까지의 과정을 살펴보자. 시스템의 부팅 절차를 알아보는 것은 여러분이 Device Driver를 설계하는데 유용할 것이다. 시스템의 부팅 절차를 이해하면 여러분이 개발한 드라이버가 언제 로드되고, 시스템의 부팅 과정 동안 여러분이 개발한 드라이버의 어떤 부분을 호출할 것인가를 결정할 수 있기 때문이다.
시스템의 부팅 절차는 시스템이나 프로세서, 운영 체제의 버전, 아키텍쳐에 종속적이기 때문에, 여기서는 시스템이 부팅되었을 때, 실제 어떤 일이 발생하는가에 대한 일반적인 정보만을 제공할 뿐이다.
시스템의 부팅 절차를 설명하는데 있어서 주요한 문제는 시작 시점을 결정하는 것이다. 여기서는 운영체제의 일부분으로써 제공되는 코드가 실행되는 시점을 시작 시점으로 정하였다. 시스템의 부팅 순서에 대해서 알아보도록 하자.

1. 시스템의 시작 모듈이 NT 시스템의 시작 루틴을 호출한다. 시작 루틴은 Boot Record 구조체를 전달 받는데, 이 Boot Record 구조체는 기본 하드웨어 정보와 나중에 운영 체제를 구성하는 Loader가 사용할 환경 정보를 담고 있다.
   NT 시스템 시작 루틴은 운영 체제에서 사용할 몇 가지 전역변수를 초기화하고 시스템이 어디에서 부팅 될 것인가에 대한 파티션과 디스크 드라이브를 결정한다. 전역 변수의 초기화 과정은 초기 시스템 부팅 단계 동안 사용할 Memory Descriptor의 초기화 하는 과정을 포함한다. 또한 시스템 시작 루틴은 Boot Loader의 Heap 영역을 초기화 하는 루틴을 호출하는데, Heap 영역을 초기화 하는 루틴은 Boot Loader가 시스템을 로딩하는 동안 메모리를 계속해서 사용할 수 있도록 적절히 Memory Descriptor를 설정한다.
   여기서 설명된 부팅 절차는 Windows NT 시스템의 부팅 절차 8단계 중 첫 번째 단계를 구성한다.

2. 시스템 시작 루틴이 Boot Loader의 시작 루틴을 호출한다. 시스템 시작 루틴은 Boot Loader 시작 루틴의 호출에 대한 응답을 기대하지 않는다. 왜냐하면 Boot Loader 시작 루틴의 호출에 대한 응답은 시스템 부팅 절차가 실패했다는 것을 나타내기 때문이다.
   Boot Loader 시작 루틴은 미리 프로세스에 의해 정의된 Boot Partition을 열고 boot.ini 파일을 읽는다. Boot Loader 시작 루틴은 boot.ini 파일을 읽기 위한 시도의 한 부분으로서 Boot Partition이 NTFS, CDFS, FAT, HPFS 파티션을 가지고 있는지 결정하기 위해 내부적으로 컴파일 된 코드를 사용한다. 이때는 기본적인 파일 시스템이 로딩되지 않은 상태이고, Boot Loader 시작 루틴은 오직 이러한 파일 시스템을 위해 지원되는(Microsoft가 부팅을 지원할 목적으로 선택) 미리 정의된 코드를 사용한다. 이것은 표준 NT File

System 구현에서 발생한다. Boot File System의 지원은 NT Boot Loader 시작 코드에서 생성하기 때문에, third-party가 부팅 가능한 File System의 구현을 제공하는 것은 Microsoft의 실질적인 지원 없이는 사실상 불가능하다. 이 시점에서 Boot Loader 시작 루틴은 video adapter를 80*50에 16-color의 alphanumeric mode로 설정하기 위해 real-mode에 있는 BIOS Interrupt를 호출한다. Real-mode에 있는 BIOS Interrupt 호출은 또한 스크린을 빈 공간으로 채움으로써 화면을 깨끗이 제거한다.

Boot Loader 시작 루틴은 boot.ini 파일의 전체 내용을 읽고, boot.ini 파일에 기록되어 있는 사용자가 사용 가능한 부팅 가능한 커널의 목록을 작성한다(멀티 운영 체제의 경우 boot.ini 파일의 [operating systems] 섹션에 부팅 가능한 운영 체제를 정의하고 있다). 파일을 읽기 위해 Boot Loader 시작 루틴은 일단 NTFS, FAT, CDFS와 HPFS 데이터 구조를 인식할 수 있는 루틴들을 다시 사용한다. Boot.ini 파일이 없다면, 설정된 기본 옵션은 Windows NT이며 부팅을 위한 기본 폴더는 C:₩WINNT이다(boot.ini 파일에서 10 entry 이상은 처리하지 못한다).

Boot Loader 시작 루틴은 boot.ini 파일의 [operating systems] 섹션으로부터 읽어 들인 옵션과 사용자가 제공한 [boot loader] 섹션에 있는 기본 부팅 위치와 일치 하는지 조사한다. Boot Loader의 시작 루틴이 기본 Boot Option과 [operation systems] 섹션에 기록된 옵션 사이에 일치하는 내용을 찾지 못하면, 선택된 기본 부팅 위치는 C:₩WINNT 이다. Boot Loader 시작 루틴이 video display 지원 루틴을 사용하는 사용자에게 기본 부팅 위치와 가능한 옵션들을 제공한다. 만약 사용자가 시스템을 부팅하는 기본 위치를 C:₩로 선택했다면, NT Loader의 시작 코드는 사용자가 DOS, Windows 3.x, Windows 95, OS/2에서 부팅 될 것을 요구한다고 생각하여 bootsect.dos 파일을 읽어 들이고, 대체 가능한 운영 체제로 시스템을 다시 부팅한다.

부팅 위치가 Windows NT이면, Boot Loader 시작 루틴은 Boot Partition의 Root Directory로부터 ntdetect.com이라는 실행 파일을 읽어 온다. 만약 ntdetect.com이 발견되지 않거나 파일의 사이즈가 잘못되었을 경우, 그렇지 않으면 운영 체제를 구성하는 Loader의 시작 코드가 만든 다른 일관성 조사가 실패했을 때, 부팅 절차는 실패하고 시스템을 다시 부팅한다. 그러나 유효한 실행 파일이 발견되었다면, 실행 파일을 메모리로 읽어 들이고, 시스템은 현재 하드웨어의 구성 정보를 찾기 위해 하드웨어 제작자가 제공하는 서비스를 사용한다.

이 시점에서 시스템 부팅 초기화 과정의 2단계로 진입한다. OS Loader의 시작 루틴은 필요한 경우 SCSI Boot Driver를 초기화 한다. 그리고 ntldr.exe OS Loader가 메모리로 로드된다.

3. OS Loader는 콘솔 입출력 장치를 열고, 시스템과 Boot Partition을 연다. 또한 OS Loader 인증 메시지인 OS Loader 4.0을 콘솔에 표시한다.

   Loader는 Windows NT 커널 시스템 이미지 파일인 ntoskrnl.exe을 위한 완전한 경로 이

름을 생성하기 위해 Boot Partition 정보를 사용한다. 시스템은 Boot Partition이 위치한 폴더 밑의 System32 폴더에서 ntoskrnl.exe 파일을 찾도록 설계되어 있다. 일단 시스템 이미지가 메모리로 로드되면, OS Loader는 hal.dll 시스템 파일을 메모리로 읽어 들인다. 이때 로드된 ntoskrnl.exe 파일과 hal.dll 시스템 파일이 사용하는 모든 DLL을 확인하고 메모리로 읽어 들인다.

OS Loader는 NT 레지스트리로부터 SYSTEM 하이브(hive)를 로딩하며, 이때는 Loader가 레지스트리로부터 LastKnownGood Control Set을 로드 할 것인지 Default Control Set을 로드 할 것인지 이미 결정된 상태이다. 이러한 결정은 Control Set이 시스템으로 로드된 Boot Driver들을 결정하기 때문에 중요하다.

메모리로 SYSTEM 하이브(hive)를 읽어오기 위해, OS Loader는 Boot Partition에 위치한 System32₩config 하이브(hive)로부터 SYSTEM 파일을 열고 이 정보를 읽는다. SYSTEM 파일의 정보 읽기가 실패한다면, SYSTEM.ALT 파일을 읽는다. 이러한 파일 읽기가 다 실패한다면, OS Loader는 부팅을 처리하지 못한다. 파일 읽기가 성공하면 파일의 내용은 유효하고, 디스크에 있는 파일 목록을 반영하기 위해 메모리에 있는 데이터 구조체들을 초기화 한다. 또한 System Loader Block(이것은 결국 로드된 시스템 이미지에 전달된다)은 메모리에 존재하는 SYSTEM 하이브(hive)의 복사본에 대한 포인터를 적절히 수정한다.

이 시점에서, OS Loader는 메모리로 읽어 들이기 위해 필요한 Boot Driver들의 목록을 결정한다. 이러한 목록에는 Boot Partition File System을 담당하는 드라이버가 포함된다. Boot Driver는 메모리에 로드된 Control Set에 있는 드라이브 키와 관련된 Start(이 값은 0으로 설정되어 있다) 값 엔트리에 의해 결정된다. 일단, Boot Driver의 목록이 결정되면, OS Loader는 이러한 드라이버들을 레지스트리에 정의된 ServiceGroupOrder에 기초하여 정렬한다. Group 내부에 있는 연속된 드라이버들은 레지스트리에 정의된 GroupOrderList와 레지스트리에 있는 각 드라이버 키와 관련된 Tag 값 엔트리에 기초를 두어 정렬된다.

드라이버 로드 순서가 결정되면 모든 Boot Driver가 로드된다. Boot Driver들을 로딩하는 동안에 발생하는 에러에 대한 이벤트는, 레지스트리에 있는 드라이버와 관련된 ErrorControl 값을 조사한다. 만약 드라이버가 시스템 부팅 절차를 위한 중요한 드라이버로 표시되어 있다면, 현재 진행중인 부팅은 실패하고, 그렇지 않을 경우 OS Loader는 다른 Boot Driver들을 로딩한다.

마지막으로 OS Loader는 로드된 시스템 이미지를 실행할 준비를 하고 NT 커널에 있는 엔트리 포인트로 제어를 전송한다.

4. 시스템 부팅 처리의 3단계에서 5단계 동안, 다양한 NT 운영 체제의 실행부(Executive)를 구성하는 요소들과 NT 커널이 초기화 된다. Start 값이 1로 설정되어 자동 로딩되는 드라이버들은 시스템 부팅 처리의 5단계 동안 로드된다.

   커널 시스템 시작 루틴은 3단계의 초기화 과정 동안 KiInitializeKernel()라 불리는 NT 커널 초기화 함수를 호출한다. 이 함수는 processor control block 데이터 구조체, 커널 데

이터 구조체, idle thread와 process object들을 초기화 하고, NT 실행부의 초기화 루틴을 호출한다. 커널 데이터 구조체와 커널 링크드 리스트(linked list) 구조체를 보호하기 위한 여러 종류의 Spin Lock이 이곳에서 초기화 된다. 또한 다양한 커널 링크드 리스트 헤더(linked list header)들(DPC 큐 리스트 헤더, 타이머 통지를 위한 리스트 헤더, 다양한 thread table list들과 다른 종류의 유사한 커널 데이터 구조) 역시 이곳에서 초기화 된다. 일단 커널 Idle thread 구조체가 초기화 되면, 실행부의 초기화 루틴이 Idle thread context에서 호출된다. NT 실행부와 이 실행부를 구성하는 다양한 구성요소들의 초기화는 2단계로 나누어서 처리된다. 실행부의 초기화 단계의 첫번째 과정 동안 다음과 같은 구성 요소들의 내부 상태를 초기화 한다.

- Hardware Abstraction Layer(HAL)
- NT 실행부 구성요소(Executive component)
- Virtual Memory Manager(VMM)

  Paged와 non-paged pool를 관리하는 Memory Manager, Page Frame Database, Page Table Entry(PTE) 관리 구조체, Mutex와 Spin Lock 데이터 구조체와 같은 다양한 VMM 자원들이 이때 초기화 된다. 또한 VMM은 시스템 Cache Working Set과 시스템 cache를 관리하기 위해 사용되는 다양한 VMM 데이터 구조체와 같은 NT 시스템 cache와 관련된 데이터 구조체들을 초기화 한다.
- NT Object Manager
- Security Subsystem
- Process Manager

  NT 실행부(Executive)의 첫번재 초기화 단계에서 초기 시스템 프로세스가 생성된다. 이때 생성된 시스템 프로세스는 초기에 생성된 Idle Process와 구분된다. 시스템 thread도 이때 초기 시스템 프로세스 context에서 생성된다. NT 실행부의 초기화 과정 중 나머지 단계에 해당하는 두번째 단계는 초기 시스템 프로세스에 속한 새롭게 생성된 thread context에서 구현된다.

NT 실행부와 이를 구성하는 다양한 구성 요소들의 두번째 초기화 단계 동안, 모든 인터럽트는 발생하지 못하도록 처리되며, 초기화를 구현하는 context에 속한 thread 권한은 가장 높은 권한으로 증가시켜 어떠한 선점(preemption)도 허락하지 않는다. 이 과정에서 시스템은 전제적인 기능을 고려하고, 실행부를 구성하는 요소들(subcomponent)은 초기화 과정을 마무리 짓기 위해 요구된 모든 동작을 구현한다. NT 실행부의 초기화 과정의 두번째 단계 동안에 다음과 같은 실행부를 구성 요소들이 호출된다.

- Hardware Abstraction Layer (HAL)가 초기화를 완료하기 위해 호출된다.
- 시스템 날짜와 시간이 초기화 된다.

- Multiprocessor 시스템에서는 다른 프로세스들이 이때 시작된다.
- Object Manager, Executive Subsystem, Security Subsystem들이 자신의 초기화 과정 중 처리하지 못한 부분을 구현하기 위해 호출된다.
- Virtual Memory Manager(VMM)의 초기화 단계 1이 구현된다.
  이때 메모리 매핑 기능이 초기화 되고, 시스템의 나머지 부분을 이용할 수 있다. VMM thread도 이때 시작된다. VMM은 전제적인 기능을 고려하고, 초기화 단계 1 이후 시스템의 나머지 부분들의 서비스를 준비한다.
- VMM의 초기화가 완료된 후 NT Cache Manager가 초기화 된다.
  Cache Manager가 초기화 되는 동안, 비동기적인(Asynchronous) 동작을 위해 요구되는 worker thread의 수가 결정되며 생성된다. 또한 Cache Manager 링크드 리스트 구조체와 동기화를 위한 자원들이 초기화 된다.
- Configuration Manager가 자신을 초기화하기 위해 호출된다.
  Configuration Manager는 NT 레지스트리를 관리한다. 초기화 단계 동안 Configuration Manager(CM)는 레지스트리에 있는 ₩REGISTRY₩MACHINE₩SYSTEM과 ₩REGISTRY₩MACHINE₩HARDWARE 하이브(hive)를 사용할 수 있도록 한다. 이렇게 함으로써 초기에 ntdetect.com가 포함한 모든 정보뿐만 아니라, OS Loader가 메모리로 가져온 정보들을 SYSTEM과 HARDWARE 하이브(hive) 내에 있는 적절한 엔트리 값으로 설정한다. 초기화 단계가 완료되면 레지스트리에 저장된 정보를 시스템을 구성하는 다른 구성 요소들이 사용할 수 있다(특히 바로 로드된 Kernel-mode Driver들은 이때 레지스트리 정보를 사용할 수 있다). 그러나 CM은 이때 수정된 사항을 레지스트리에 저장하지는 못한다. 드라이버에 한정된 초기화를 구현하기 위해 요청된 Kernel-mode Driver들은 레지스트리에 등록된 정보들을 액세스하기 위하여 표준 레지스트리 루틴을 사용할 수 있다.
- NT I/O Manager가 자신의 초기화 작업을 구현하기 위해 호출된다.
  I/O Manager는 먼저 동기화 데이터 구조체, 링크드 리스트, 메모리 저장 공간(특히 IRP Zone/Lookaside List)를 포함하는 지정된 개체(state object)들을 초기화 한다. 그 다음 I/O Manager는 ObCreateObjectType() 내부 함수를 사용하는 Object Manager를 호출하여 내부적으로 정의된 모든 Object 타입을 등록한다. 내부적으로 정의된 Object의 종류에는 Adapter Objects, Controller Objects, Device Objects, Driver Objects, I/O Completion Objects와 File Objects들이 포함된다. 이때 I/O Manager는 Object의 이름을 관리하는 영역(name space)에 ₩Device, ₩DosDevices, ₩Driver 루트 폴더를 생성한다.
  그 다음으로, I/O Manager는 OS Loader가 메모리에 로드한 Boot Driver들을 초기화 한다. 이것은 드라이버에 한정된 초기화를 구현하기 위해 로드된 드라이버들 각각의 Driver Entry Routine이 호출된다. 이때 Raw File System Driver도 로드된다. Raw File System Driver 이전에 로드된 다른 File System Driver는 Boot File System Driver가 유

일하다. 드라이버들은 NT 레지스트리와 상호 동작하는 기능을 제안해야 한다. 마지막으로 Start 엔트리 값이 1로 설정된 드라이버들이 로드되고, 드라이버에 한정된 초기화를 위해 드라이버가 가지고 있는 Driver Entry Routine이 호출된다.

재 초기화 작업을 요구하는 모든 로드된 드라이버를 위해 드라이버의 재 초기화 루틴이 이어서 호출된다. 드라이버의 재 초기화 작업 후에 NT I/O Manager는 Disk Partition을 인식하기 위해 드라이브 문자를 할당한다. I/O Manager는 CD-ROM 드라이브나 하드디스크 드라이브 파티션을 보호하기 위해 고정으로 할당한 드라이브 문자가 있는지 레지스트리를 검사한다. 이러한 드라이버 문자의 할당은 동적으로 DOS 드라이버 문자를 할당할 때 사용할 수 없도록 고정되어 있다.

각각의 하드디스크를 위해 드라이브 문자를 할당하기 전에 I/O Manager는 물리 드라이브에 대한 Open 동작을 구현한다. 만약, 여러분이 하드디스크 Device Driver나 Lower-level Filter Driver 또는 Intermediate Driver를 개발한다면, 개발자는 이 시점에서 Open 요청을 처리해야 한다. Open 요청이 성공하면 Device Object에 대한 Symbolic Link가 NT Object 이름을 관리하는 영역(name space) 내부에 생성되며, 이 링크에 할당된 이름은 ₩DosDevices₩PhysicalDrive%d의 형식을 가진다. 여기서 %d는 순차적으로 할당된 디스크 드라이브 숫자이다. NT I/O Manager는 이 시점에서 디스크 드라이브로부터 파티션 정보를 요구한다. 고정된 Disk Partition에 드라이브 문자를 할당하기 위한 순서를 결정하기 위해 I/O Manager는 다음과 같은 방법을 사용한다.

- NT I/O Manager는 보호하기 위한 고정된 드라이브 문자가 할당되어 있는지 레지스트리에 요청한다.
- 부팅 가능한 파티션이 먼저 DOS에서 사용 가능한 드라이브 문자를 동적으로 할당한다(특히 파티션을 대표하는 Device Object를 위해 ₩DosDevices₩%c라는 이름을 가진 Symbolic Link를 생성한다. 여기서 %c는 파티션을 위해 NT I/O Manager가 선택한 드라이브 문자를 나타낸다).
- 다음으로 기본 파티션(Primary partition)이 동적으로 드라이브 문자를 할당하기 위해 선택된다.
- 확장 파티션(Extended partition)이 이어서 DOS 드라이브 문자에 할당된다.
- 다른 파티션에 DOS 드라이브 문자들을 할당한다. 하드디스크 드라이브와 Removable Driver Partition을 위해 드라이브 문자가 할당된 후, NT I/O Manager는 하드웨어를 검색하는 동안에 인식된 모든 CD-ROM 드라이브들에 대한 드라이브 문자를 할당한다.
- Local Procedure Call(LPC) Subsystem과 Process Manager Subsystem이 지금 자신의 초기화 과정을 완료한다.
- 다음으로 Reference Monitor와 Session Manager Subsystem이 자신의 초기화 과정을 완료하기 위해 호출된다.

5. 여기까지 시스템의 부팅 처리의 3단계에서 5단계까지 마무리 되었다. 여기서 기억해야 할 것은 NT 실행부(Executive)를 구성하는 요소(component)들은 NT 커널이 생성한 시스템 프로세스에 속하는 시스템 worker thread의 context에서 초기화 되었다는 사실이다. 시스템 worker thread는 zero page thread라는 Memory Manager의 역할을 수행한다. Zero page thread는 VMM이 Free List에 위치한 페이지들을 비동기적인 방식으로 0으로 제거하는데(Zero Out) 사용되는 가장 낮은 권한을 가지고 있는 thread이다. 모든 페이지는 시스템이 US Department Of Defense(DOD)가 정의한 C2 보안을 확인하기 위해 페이지를 다시 사용하기 전에 0으로 제거하는(zeroed out) 것이 필요하다.

6. 이 시점에서 시스템의 초기화 완료 되었다. 시스템 부팅 절차의 6단계에서 8단계 동안 다양한 다른 구성 요소(subsystem)들이 초기화 되고 Service Controller Manager가 다른 서비스들을 로드한다. 여기에는 레지스트리에 있는 Start 값이 2로 설정된 Kernel-mode Driver의 로딩이 포함된다.