

< 논문 >

운영체제(Operating System) 의 역사와 장래

한국 해양대학교 해사수송학과 4학년 홍성일
담당 교수 : 정태권 교수님



1. 서론
 2. 운영 체제란 무엇인가?
 3. 운영 체제의 종류와 발전사
 - 가. DOS
 - 나. UNIX
 - 다. WINDOWS NT
 - 라. OS/2
 - 마. NEXT STEP
 - 바. LINUX
 4. 결론
- * 참고 문헌

1. 서론

본 논문에서는 현재 컴퓨터에서 쓰이고 있는 운영 체제(Operating System)에 대해 그 중요성을 알아보고 발전사를 바탕으로 앞으로 나올 차세대 운영 체제에 대하여 논술함이 목적이다. 많은 시스템에서의 운영 체제가 있지만, 개인용 컴퓨터(Personal Computer)에서 쓰이는 운영 체제 위주로 쓰여질 것이다. 그리고, 아직 앞으로 나올 운영 체제에 대해 논란이 많지만, 본인의 생각과 자료를 바탕으로 차세대 운영 체제들을 논술하겠다.

2. 운영 체제란 무엇인가?

흔히 컴퓨터라 하면 눈에 보이는 물리적 기계(Hard Ware)만을 생각하기 쉽다. 그러나, 실제로 그런 기계를 움직이는 것은 눈에 보이지 않는 프로그램(Soft Ware)이라는 것을 대부분 알고 있다. 이러한 소프트웨어 중에서 가장 중요한 역할을 하는 것이 바로 운영 체제(Operating System)이다.¹⁾ 이 운영 체제는 사용자가 자신의 컴퓨터를 사용할 수 있게 사용자, 하드웨어, 소프트웨어 사이에 인터페이스(Interface)를 제공한다. 즉, 이 말은 프로그램의 실행, 기억 및 처리장치의 관리 등의 기능을 수행하는데 인간의 간섭없이 직접 컴퓨터를 운영한다는 말이다. 운영 시스템은 여러 가지 기능을 수행하는데 그 중 가장 중요한 기능 두가지는 컴퓨터를 출발시키는 것, 즉 초기 프로그램 적재 또는 부트스트랩을 수행하는 것과 각 작업의 처리를 시작하는 것이다.

쉽게 말해 운영 체제는 컴퓨터라는 기계가 어떻게 동작해야 하는지를 규정해 놓은 법률서적과 같은 것이다. 컴퓨터 내부에 있는 모든 개별적인 요소들(프로세서, 메인 메모리, 하드 디스크 등)은 운영 체제의 통제를 받아서 동작하게 된다.

간단히 운영 체제의 역사를 서술하면 초창기 컴퓨터에는 운영 체제라는 것이 존재하지 않았다. 당시의 컴퓨터는 단순히 수치 계산이라는 한가지 목적으로 만들어 졌기 때문에 운영 체제를 생각하지도 않았던 것이다. 그러나, 컴퓨터가 점점 넓은 범위에서 활용되기 시작하면서 1950년대 작업과 작업간의 원활한 유통을 위해 처음으로 운영 체제가 도입되기 시작했다. 이때의 운영 체제는 일괄 처리방식으로서 하나의 작업이 실행중일 때는 모든 제어기능을 실행중인 작업이 갖고 있지만, 작업이 끝나면 제어 기능이 운영 체제로 넘겨져서 다른 작업을 수행할 수 있도록 한 것이다. 1960년대에는 다중 처리작업, 다중 처리, 시분할, 실시간 처리 등을 가능하게 하는 운영 체제가 계속적으로 도입되었고, 1970년대에 가서는 광범위 컴퓨터 네트워크(Net Work)간의 통신이나 온라인 처리 등을 관리하는 운영 체제가 나타났으며 특히, 개인용 컴퓨터를 위한 운영 체제가 많이 개발되었다.

3. 운영 체제의 종류와 발전사

가. DOS(Disk Operating System)

²⁾DOS(Disk Operating System)는 초창기의 IBM PC에서 사용하기 위한 목적으로

IBM과 마이크로소프트(Microsoft)의 공동노력으로 개발된 것이다. 최근 10여년 동안 IBM 개인용 컴퓨터와 호환 기종이 현재의 386이나 486기종으로 개선되어 왔듯이, DOS도 그러한 컴퓨터에 맞게 개정되어 왔다. DOS는 세가지 기능을 제공하는데 우선, 메모리나 디스크 드라이버와 같은 시스템 자체를 제공한다. 다음으로, 사용자가 사용할 수 있도록 인터페이스를 제공한다. 마지막으로, DOS는 사용자가 시스템을 효율적으로 관리할 수 있도록 도와준다.

사용자는 두가지 사용자 인터페이스 중 하나를 선택할 수 있다. 일반적으로 전통적인 DOS명령 프롬프트와 새로운 DOS Shell이 있다.

▶ 명령 프롬프트

명령 프롬프트(Command Prompt)는 DOS 1 이후에 계속 사용되어 왔으며 여러 가지 장점을 제공하지만 많은 단점을 가지고 있다. 그러나, 많은 사람들은 여전히 DOS 명령 프롬프트를 통해서 작업을 수행하는 경향이 있다. DOS는 “나는 당신의 다음 명령을 수행하기 위한 준비가 되어 있습니다.”라고 알려주는 간단한 메시지인 명령 프롬프트 'C:\>'를 출력한다. 대부분의 사용자에게 있어서 명령 프롬프트의 단점은 명령을 입력하는 것 자체에 있다. 소프트웨어의 유용성은 종종 얼마나 자유로운 상태에서 그것을 사용할 수 있는지의 정도와 DOS의 표준 명령 프롬프트가 제한된 형식만을 받아들인다는 전통성을 비교하여 상대적으로 측정된다. 사용자는 DOS 명령 프롬프트에서 제한된 명령과 엄격한 문법을 지키며 매년 일정한 형식으로 명령을 입력해야 한다.

▶ DOS Shell

최근 키보드보다 마우스를 더욱 필요로 하는 그래픽 사용자 인터페이스(Graphic User Interface)는 많이 성장했다. GUI 환경에서는 직접 명령을 입력하는 대신에 메뉴를 통해서 명령을 선택할 수 있다. 사용자는 디렉토리를 확인하고, 사용하기 원하는 파일을 클릭할 수 있다. 파일을 삭제할 때에는 선택된 파일이 목록에서 사라지는 것을 직접 눈으로 확인할 수 있다. 파일의 이동시키거나 복사할 때에는 화면상에 그에 대한 결과가 출력된다. 심지어 많은 동작은 메뉴를 사용하지 않는 상태에서도 수행될 수 있다. 또한, 사용자는 작업을 수행하기 위해서 필요한 파일을 화면상의 다른 곳으로 끌어낼 수 있다. 이러한 방법은 대부분의 사람들에게 있어서 엄격한 문법과 제한된 명령을 입력하는 것보다 더욱 자연스럽게 편리하다. DOS는 DOS 4에서부터 시작하여 DOS Shell이라고 하는 그 자신의 그래픽 사용자 인터페이스를 제공했다. Shell은 DOS 5에서 상당히 개선되었다. DOS 6에서는 크게 달라진 사항은 없다. 명령 프롬프트에서는 단지 한 번에 하나의 명령만을 실행할 수 있다. 메모리 상주 프로그램의 실행이 끝날 때까지, 명령 프롬프트는 다시 출력되지 않는다. 그러나, DOS Shell에서는 여러개의 프로그램을 실행시킨 상태에서 그들 간에 전환하며 각각의 프로그램을 사용할 수 있다. 이러한 방법을 작업 전환(task swapping)이라고 한다.

다음은 DOS의 기능에 관한 것으로 사용자를 돕기 위해 다음과 같은 기능이 제공되어진다.

▶ 명령의 입력에 관련된 기능

- DOSKEY 프로그램은 모든 명령을 반복해서 입력하지 않아도 이전의 명령을 다시 사용할 수 있도록 명령을 기록,재생,수정하게 해준다. 또한, 이 프로그램은 복잡한 명령을 쉽고 간단한 형식으로 대치하여 사용하게 해준다.
- 화면상에 출력되는 도움말 기능을 사용하여 명령의 형식과 사용 방법에 대한 내용을 참고할 수 있다.
- 기호에 맞게 명령 프롬프트를 직접 제작할 수 있다.
- 만약 명령의 결과가 한 화면을 벗어난다면, 그것을 여러 페이지로 나누어 출력할 수 있다.
- 화면에 출력되는 명령의 결과나 메시지를 파일이나 프린터,또는 다른 어떤 곳으로 전달할 수 있다. 심지어 그러한 출력 내용을 다른 명령의 입력 내용으로 사용할 수도 있다.
- 하나의 명령을 여러 파일에 대하여 반복하여 사용할 수 있다.

▶ 디스크와 데이터를 관리하는 기능

DOS의 주요 기능의 하나는 모든 프로그램에서 디스크상에 동일한 방법으로 데이터를 저장하고 그것을 사용하도록 디스크를 관리하는 것이다.DOS에서 실행되는 어떤 소프트웨어는 디스크에서 데이터를 직접 읽어들이거나 그곳에 저장하지 않는다. 그러한 소프트웨어는 DOS에게 그것을 수행하도록 지시한다. 대부분의 경우, 사용자는 DOS가 이러한 동작을 수행하는지 신경 쓸 필요가 없다. 이러한 동작은 우리가 인식하지 못하는 상태에서 수행된다.

컴퓨터를 사용하는 대부분의 사용자들은 문서 작성기나 데이터베이스 관리 프로그램과 같이 자신이 사용하기 원하는 응용 프로그램을 통해서 디스크상의 파일을 사용하게 된다.그러나, DOS는 이러한 대부분의 응용 프로그램이 제공하지 않는 많은 유용한 기능을 제공한다.

- DOS는 디스크를 관리하는 여러 가지 명령을 제공한다. 예를 들어, FORMAT 프로그램은 새로운 디스크를 사용할 수 있는 상태로 준비해 주며, DISKCOPY 프로그램은 플로피 디스크를 복사하여 백업 복사본을 만들어준다.
- DOS는 디스크상에서 파일이 저장되어 있는 곳을 구분하기 위해서 디렉토리 체계를 사용한다. 사용자는 디렉토리를 생성하는 MD, 디렉토리 목록을 출력하는 DIR, 디렉토리를 RD와 같은 명령을 사용하여 디렉토리를 관리할 수 있다.
- DOS의 대부분의 명령은 사용자가 파일을 관리하도록 도와주는 것들이다. 사용자는 파일을 복사하고 내용을 살펴보고 그것을 인쇄하고 정렬할 수 있다. EDIT라는 이름의 에디트는 간단한 파일을 생성하고 수정하게 해준다.
- 용량이 큰 하드 디스크에서 생각보다 자주 발생하게 되는 특정 파일의 위치를

잊어버리는 경우에 사용자는 파일의 이름이나 그것의 내용을 생성하고 수정하게 해준다.

▶ 데이터를 보호하고 복구하는 기능

아주 많은 사람들이 과거 10여년 전부터 개인용 컴퓨터를 사용해 왔으며, 사용자들은 여러 가지 원인으로 데이터를 손실하거나 잃어버렸다. 실수로 파일을 삭제하는 것과 같이 아주 쉽게 발생하는 원인이 있는가하면, 다른 어떤 원인은 정말 골칫거리가 되기도 한다. 또한, 컴퓨터를 사용하는 사람의 실수가 데이터를 잃게되는 유일한 원인이 되지 않는다. 하드웨어 자체와 저장 매체상의 결함도 대부분의 문제에서 원인이 될 수 있으며, 특히 컴퓨터 바이러스는 더욱 악명을 떨치며 증가하고 있다. DOS의 새로운 버전이 나타날 때마다, DOS는 이러한 데이터의 손실에 대비하여 사용자의 데이터를 보호할 수 있도록 도와주며 필요하다면 그거을 복구하게 해주는 몇가지 새로운 기능을 제공하였다.

- DOS는 파일을 삭제하고 나서 며칠 후에 그것을 다시 복구하게 해주기 위해서 특정 기간동안 삭제된 파일을 보존해 두는 기능을 제공한다.
- 마이크로소프트의 Anti-Virus는 컴퓨터 바이러스가 어떤 파괴적인 동작을 수행하기 전에 그것을 미리 검색하여 제거한다.
- 모든 디스크는 그것을 사용하기 전에 초기화 되어야 하지만, 가끔 사용자는 이미 유용한 데이터가 들어있는 디스크를 실수로 초기화하게 된다. DOS는 이러한 디스크의 초기화 상태를 복구시켜준다.
- 파일이 변경되거나 삭제되지 않도록 방지하기 위해서 각각의 파일을 쓰기 방지 상태로 만들 수 있다.
- 다른 사람이 사용자의 디렉토리 목록 내의 파일을 살펴볼 수 없도록 그것을 숨길 수 있다.
- 만약 좋지 못한 데이터가 디스크상의 디렉토리와 시스템 자료에 포함되면 DOS는 파일이 완전하더라도 그것을 발견할 수 없게 될 것이다. Check Disk 프로그램은 그러한 문제를 찾아서 정정할 수 있다.
- 모든 데이터 손실을 완벽하게 방지할 수는 없으므로, DOS는 사용자가 무엇보다도 중요하고 유용한 파일의 백업 복사본을 쉽게 생성할 수 있도록 도와준다. 만약 이러한 백업 복사본을 정기적으로 생성한다면, 사용자는 컴퓨터를 도난당하거나 그것이 불에 타버릴지라도 데이터를 완벽하게 복구할 수 있을 것이다.

▶ 메모리를 관리하는 기능

② 메모리(Memory)는 DOS가 항상 내부적으로 동작을 수행할 수 있는 다른 하나의 영역이다. DOS는 메모리중에서 가장 낮은 부분에 위치되어 있고 사용자의 프로그램이 실행되는 상용 메모리(Conventional Memory)의 사용을 제어한다. 또한, DOS는 286이나 상위 기종에서 메모리 내의 다른 영역을 제어할 수도 있다. DOS

는 다음과 같은 동작을 수행하는 명령이나 기능을 제공한다.

- 상용 메모리가 아닌 다른 메모리 영역을 관리한다.
- 시스템에서 사용하기 위한 목적으로 보존되어 있던 메모리 영역으로 소프트웨어를 읽어들인다. 이것은 상용 메모리 공간을 많이 절약하게 해준다.
- 현재 사용중인 메모리의 용도와 유효한 영역에 대한 보고서를 출력한다.
- 시스템의 메모리를 취대로 활용하기 위해서 시스템을 최적화 한다.

▶ 시스템의 효율을 개선하는 기능

DOS는 사용자의 시스템이 더욱 빠르게 실행되도록 해주고, 사용자가 더 많은 디스크 공간을 사용할 수 있게 해주는 여러 가지 기능을 제공한다.

- DOS의 DEFRAG 프로그램은 가능한한 가장 빠르게 디스크를 사용할 수 있도록 디스크상의 데이터를 정리한다. 사용자는 항상 하드 디스크를 정리된 상태로 유지하기 위해서 정기적으로 DEFRAG와 같은 프로그램을 실행해야 한다.
- 사용자는 실제 드라이브 보다 더욱 빠른 속도로 동작하는 램 드라이브(RAM-drive)를 생성할 수 있다. 램 드라이브는 실제적으로 어떤 물리적인 장치를 동작시키지 않고 메모리 영역을 사용하여 디스크 드라이브를 모방하는 가상적인 드라이브 형태이다.
- DOS의 Double Space 프로그램은 일반적인 하드디스크에 지금보다 두배이상의 많은 데이터를 저장할 수 있도록 디스크상에 데이터를 압축하여 저장한다.
- DOS의 Smart Drive 프로그램은 특정 메모리 영역에 디스크상의 데이터를 복사하여, DOS가 디스크 대신에 메모리에서 데이터를 읽어들이거나 그곳에 기록한다. 이러한 방법은 상대적으로 느린 디스크 드라이브를 사용하지 않고 필요할 때마다 즉시 프로그램을 읽어들이 수 있으므로, 사용자는 프로그램을 더욱 빠르게 실행할 수 있다.

▶ 배치 파일을 수행하는 기능

DOS는 여러 가지 DOS 명령을 결합시켜 배치 프로그램(Batch Programs)이라는 사용자만의 독특한 프로그램을 생성하게 해준다. 사용자는 배치 프로그램을 작성하기 위해서 프로그래머와 같이 복잡하고 많은 내용을 이해할 필요가 없다. 실제적인 프로그래밍에 대해 전혀 모르는 많은 초보자들도 배치 프로그램을 쉽게 작성할 수 있다.

사용자는 배치 프로그램 내에서 모든 명령을 사용할 수 있으며, DOS는 여러 가지 특수한 명령을 제공해 준다.

- 사용자는 AUTOEXEC.BAT 파일이 존재하는지 확인하는 것과 같이, 특정 조건의 결과에 따라 그 다음에 실행하는 명령을 결정할 수 있다.
- 사용자는 다른 사람을 위한 독특한 메시지를 출력할 수 있다.

- 사용자는 키보드에서 문자를 읽어들이고, 입력된 내용에 따라 다음으로 수행하는 것을 결정할 수 있다.

▶ 하드웨어를 관리하는 기능

또한, DOS는 사용자의 하드웨어를 직접 관리하고 제어한다. 많은 응용 프로그램은 키보드, 모니터, 프린터 등을 직접 사용하지 않고, DOS를 통해서 그것을 사용한다. 그러나, 이것은 모든 경우에 있어서 적용되는 사실은 아니다.

DOS는 하드웨어의 주변장치를 실제로 제어하는 프로그램인 디바이스 드라이버(Device Drivers)를 통해 하드웨어를 관리한다. 예를 들어, 프로그램에서 프린터상의 한줄의 데이터를 인쇄할 필요가 있을 때, 프린터 드라이버는 프린터 헤드를 위치시키고 폰트를 선택하고 인쇄하는 것과 같은 동작을 수행한다. 디바이스 드라이버는 DOS에서 사용할 수 있는 하드웨어의 종류에 있어서 더욱 융통성을 제공한다. 만약 하드웨어를 관리하는 기능이 DOS 내부에만 내장되어 있다면, 운영체제는 이미 그곳에 내장되어 있는 프린터, 모니터, 키보드 등만을 다룰 수 있을 것이다. 그러나 드라이버를 사용할 경우, DOS는 드라이버를 포함하는 모든 하드웨어를 사용할 수 있게 해준다. DOS는 키보드, 프린터, 모니터에 기본적인 드라이버를 제공한다. DOS는 다음과 같은 동작을 수행하게 해준다.

- 부팅 과정에서 디바이스 드라이버를 설치한다.
- 직렬 포트에서, 전송율과 같은 통신 포트의 상태를 설정한다.
- 사용자의 하드웨어가 특수한 문자를 출력하거나 인쇄할 수 있도록 설정한다.
- 사용자와 DOS 사이에 데이터나 자료를 전달하기 위해서 키보드와 모니터가 아닌 다른 어떤 입출력 장치를 사용한다.
- 시스템의 효율을 최대한 높이기 위해서 다양한 하드웨어의 설정 상태를 최상의 상태로 조절한다.

나. UNIX

유닉스 시스템의 기원은 1960년대 후반 MIT와 AT&T사의 벨(BELL)연구소 및 당시의 컴퓨터 제작회사인 GE(General Electric)사가 MULTICS라 불리는 실험적 운영 체제에 관해서 연구했을 때부터 시작된다고 할 수 있다. 다른 송신 정보 컴퓨터 활용 시스템(MULTIplexed Information and Computer System)으로 부터 따온 MULTICS는 보안을 유지하면서도 정보를 배분해주는 GE645 대형 컴퓨터에 대해 상호 작용 운영 체제가 되도록 고안되었지만 개발시간이 많이 지체되었고, 생산된 버전들은 속도가 느리고 광대한 메모리를 필요로 했기 때문에 벨 연구소는 그 계획을 취소하지 않을 수 없었다. 그러나 MULTICS는 여러 가지 혁신적인 특징들을 수행했고 뛰어난 컴퓨터 활용 상황을 창출하게 되었다.

1969년 벨 연구소의 MULTICS 계획 수행자 중 연구원 Ken Thompson은 'Space Travel'이라는 GE 컴퓨터를 위한 게임을 만들었는데, 이 게임은 태양과 우주선을 본뜬 것이었다. 그러나, Thompson은 이 게임이 급격하게 GE기계를 작동시키고

비용이 많이 든다(한 게임당 75달러)는 것을 알고는 다시 Dennis Ritchie의 도움으로 그 게임을 DEC에서 만든 PDP-7의 여백에 실기 위해서 다시 만들게 된 것이다. 이러한 초기의 경험은 그로 하여금 Thompson과 Ritchie, Rudd Caunday가 함께 고안한 파일 시스템 구조를 사용하여 PDP-7에 고안한 새로운 운영 시스템을 만들 기회를 주게 된 것이다. Thompson과 Ritchie와 그들의 동료들은 파일 시스템과 명령 해독기 기능 그리고 PDP-7을 위한 몇몇 유틸리티를 포함한 다중 임무 수행 운영 시스템을 개발 하였으며, 이것으로 두 사용자가 동시에 사용할 수 있게 되었고, 그것은 UNiplexed Information and Computing System(UNICS)라는 이름으로 불리게 되었다. 이 이름은 1970년대에 들어와서는 UNIX로 약간 바뀌었다.

그리고 1973년에 Ritchie와 Thompson은 C프로그래밍 언어로 커널(kernel)을 다시 사용하여 대규모 시스템에서도 사용할 수 있게 하였다. 1974년까지 UNIX시스템은 벨 연구소에서 활발하게 사용되었으며, 1977년까지는 여러 가지 새로운 툴(tool)과 유틸리티들을 포함하고 있었다. UNIX시스템을 가동하는 기계의 수는 주로 벨 연구소와 대학에서만도 1978년까지는 600대 이상으로 증가하였고, 1979년에 나온 UNIX는 현재 사용되고 있는 UNIX의 전신이라고 할 수 있다.

UNIX시스템III는 1979년 판을 근거로 해서 만들었고, 1982년 AT&T사는 계열회사인 Western Electric사를 통해 UNIX시스템의 버전들을 계속해서 판매 하였다.

▶ UNIX시스템의 구성

■ UNIX시스템V

① UNIX시스템의 다양성에 대한 여러 가지 혼란을 정리하기 위해서 AT&T사는 1983년에 UNIX시스템V Release1을 도입하였다. UNIX시스템V R1과 함께 처음으로 AT&T사는 앞으로의 UNIX시스템 Release에 '계속적 겸용 사용 기능'을 유지할 것을 약속했는데 이것은 R1에 설계된 프로그램들이 시스템V의 앞으로 나올 Release로도 작동을 계속할 수 있을 것이라는 것을 의미한다.

UNIX시스템 버전으로부터 몇 가지 특징들을 결합한 R1은 화면 편집기 vi와 화면조정 라이브러리인 Curses와 함께 버클리 대학에서 개발되었다. AT&T사는 1985년 유닉스 시스템V R2를 개발하여 도입하였으며, R2는 정전과 전기 충격시에의 파일 보호와 프로그램, 작업 조정 특징들과 향상된 시스템 관리 등에 대한 독점적 사용을 위해 파일과 데이터들을 잠그는 것 등을 도입했다. 1987년 AT&T사는 유닉스 시스템V R3을 도입했고 그것은 통신망 연결에의 단순하고 지속적인 접근법이 포함되어 있다. 이 시스템들은 "스트림(stream)"이라는 통신망 연결 소프트웨어를 만들기 위해 사용되는 것과 통신망을 따라 파일을 분배하기 위한 장거리 파일 시스템, 통신망 연결에 이용되는 응용 프로그램을 만들기 위해 사용되는 Transport Level Interface(TLI) 등을 포함하고 있다.

R3.1은 보다 많은 문자를 지정하고 날짜, 시간 지정을 함으로써 유닉스 시스템은 전세계적으로 사용할 수 있도록 만들었다. 이것은 또한 메모리 사용과 파일의 백업과 회복을 위한 몇몇 중요한 기능, 수행, 증진도 고려되었다.

R3.2는 보다 증진된 시스템 보안이 고려되었는데 사용자의 최근login 시간을 보

여주고 실패한 login 시도를 기록하며 사용자들이 암호화된 암호명(password)을 읽지 못하도록 숨겨진 암호명을 기록하는 등의 기능을 하였다. R3.2는 Framed Access Command Environment(FACE)를 도입하기도 했는데, 그것은 메뉴로 작성된 사용자 고유 영역을 제공한다. SVR 4는 AT&T사 자체 내에서나 외에서 개발된 유닉스 시스템의 여러 다양한 버전들을 종합하고 있다.

■ BSD(Berkeley Software Distribution)

① 유닉스 시스템에 관한 여러 가지 중요한 혁신들은 캘리포니아 버클리 대학에서 이루어졌는데, 이러한 발전의 몇몇은 유닉스 시스템V 초기 Release들의 일부가 되었고 그 외의 많은 부분을 R4에 도입했다. 버클리 대학은 1974년 제4판을 시작하여 유닉스 시스템에 관여하게 되었는데 유닉스 시스템 버클리 이전의 개발은 1975년 Ken Thompson이 컴퓨터 과학부에서 1년간 연구 휴가를 얻었을 때부터 시작된다. 버클리 대학에 있을 동안 Ken Thompson은 많은 사용자에게 유닉스 시스템을 사용가능하게 한 PDP-11/70의 제6판 연구에 착수했으며, 대학원생이었던 Bill Joy와 Chuck Haley는 버클리 버전에 큰 공헌을 했다. 그들은 편집기를 ed라 불리우는 것으로 통합했고 파스칼 부호 번역기를 개발했다. Joy는 그가 'Berkeley Software Distribution'이라고 명명한 패키지를 종합했다. 그는 또한 여러 다른 중요한 혁신을 이루었는데 그 중에는 C셀과 화면 작성 편집기 vi도 포함된다.

1978년 두 번째 버클리 버전이 만들어졌는데 이것은 2BSD로 약칭된다. 1979년에는 3BSD가 배분되었다. 그것은 2BSD와 제7판을 근거로 두는데 작동시킬 수 있는 메모리 보다 더 큰 프로그램을 만들 수 있도록 하는 실제적 메모리 특징을 가지고 있다.

3BSD는 DEC VEX-11/780을 작동시키기 위해 발명되었다. 1970년 후반, 미국방부 산하 첨단 연구계획 기관(DARPA)은 유닉스 시스템에 그들의 다방면에 걸친 컴퓨터 활용 상황을 기초로 둘 것을 결정했다. DARPA는 그들 자신의 유닉스 시스템 버전 개발을 버클리 대학에서 실시할 것을 결정했고, 결과적으로 DARPA는 4BSD 개발을 위해 기금을 주었다.

1983년에는 4.1BSD가 나왔는데 그것은 기능 수행 능력이 향상된 것이다. 또 1983년에 나온 4.2BSD는 파일 대체시킴과 원거리 login을 위해 사용될 수 있는 TCP/IP 통신망 연결과 파일에서의 접근속도를 빠르게 한 새로운 파일 시스템을 포함한 통신망 연결 특수 기능을 도입했다. R 4.3BSD는 4.2BSD를 약간 변형시켜 나온 것이다.

많은 컴퓨터 판매자들은 BSD 시스템을 그들의 유닉스 시스템 개발을 위한 기초로서 사용했다. 이 변형들 중 가장 중요한 것은 Joy가 공동 설립한 선 마이크로 시스템(SUN micro system)에 의해 개발된 선 운영 시스템(SUN OS)이다. SUN OS는 통신망 파일 시스템과 통신망 연결 특수 기능을 포함하여 4.2BSD에 여러 기능을 덧붙인 것이다. SUN OS는 유닉스 시스템V Release 4.0에 합쳐진 구성 요소 중의 하나이다.

▶ 유닉스의 특징

유닉스의 특징은 한마디로 멀티 유저(multi user)와 멀티 태스킹(multitasking)이 가능한 운영 체제라는 것이다. 유닉스를 사용하면 한 대의 컴퓨터를 동시에 여러명이 사용할 수 있으며, 여러 작업(process)을 동시에 처리할 수 있다. 이것은 여러 사용자가 파일이나 정보를 공유할 수 있게 하므로 작업하기에 편리한 환경을 제공해 주며, 동일한 데이터를 중복해서 저장하지 않아도 되므로 비용의 절감을 기대할 수 있다.

유닉스는 대화식(interactive)운영 체제이다. 즉, 사용자가 어떤 명령을 내리면 시스템은 명령을 수행하고 적당한 응답을 보내준다. 그 다음에 다시 사용자가 명령을 내리면 시스템은 명령을 수행하고 다시 응답하는 방식으로 작업을 진행시킨다. 이러한 특징은 사용자가 유닉스를 편리하게 사용하도록 하며, 동시에 문제점을 금방 파악할 수 있도록 하여 프로그램 개발 기간을 단축시켜 준다.

한편 유닉스는 내장 매뉴얼을 제공하고 있다. 따라서 사용자가 특정 명령의 사용법을 모를 경우에 온라인 상으로 사용법에 대한 설명을 찾아 볼 수 있다. 그러나, 작은 유닉스 시스템에는 매뉴얼이 없는 경우도 있다.

또 유닉스는 셸(shell)이라는 사용자 인터페이스를 제공한다. 셸은 일종의 명령어 해석기(interpreter)를 말하는데, 이것은 사용자가 입력하는 명령어를 해석, 수행하는 프로그램으로 도스의 COMMAND.COM과 같은 것이다. 유닉스에서 셸은 각종 명령의 처리뿐 아니라, 훌륭한 프로그래밍 언어의 역할도 한다. 따라서 사용자들은 포트란(FORTRAN)이 C와 같은 고급 프로그래밍 언어를 전혀 몰라도 셸을 사용하여 쉽게 처리할 수 있다. 이러한 프로그램은 도스의 배치파일(batch file)과 같은 것이라고 말할 수 있으나, 도스의 배치파일은 유닉스 셸에 비하면 초보적인 수준에 불과하다.

유닉스는 본래 프로그래밍에 관한 연구를 좀 더 편리하고 효율적으로 수행하기 위해 만들어진 운영 체제이다. 따라서 유닉스는 에디터와 C, LISP, SNOBOL, 포트란, 파스칼, 베이직 등과 같은 프로그래밍 언어는 물론 여러 가지 유용한 유틸리티를 제공하고 있다. 이러한 유틸리티 가운데 텍스트 조작을 위한 것으로 ed, vi, sed, grep, tr, rpl, awk, sort, uniq 등이 있으며, 문서를 일정한 양식으로 포맷팅하기 위한 nroff, troff, eqn, tbl 등을 구비하고 있다. 또 계산을 편리하게 도와주는 bc, dc 등과 같은 대화형 계산기 프로그램도 있다. 이밖에 프로그래밍 작업을 도와주는 make, lex, yacc 등과 같은 유틸리티들도 제공한다. 특히 nroff, eqn는 사용자가 작성한 프로그램의 설명문을 작성하도록 제공한 것이지만, 기능이 매우 강력하여 간단한 도표는 물론 수식까지도 표현이 가능하므로 전문가가 아니더라도 많이 애용하고 있다. 한편, 유닉스는 매우 독특한 파일 시스템을 가지고 있기 때문에 흔히 계층적 트리(tree)구조라고 말하는데, 여기서 계층적 구조란 유닉스 시스템이 디렉토리라는 특수 파일을 가지고 있음을 의미한다. 디렉토리라는 파일은 사용자의 데이터를 가지는 것이 아니라 파일의 목록과 파일 시스템에서의 파일 위치에 대한 정보를 가지게 된다.

유닉스의 특징을 설명하면서 빼놓을 수 없는 것이 강력한 통신 기능이다. 전화

선이나 전용선을 사용하여 다른 시스템의 파일을 복사해올 수도 있고, 또다른 시스템을 연결하여 사용할 수도 있으며, 같은 시스템 내의 사용자들과 메시지를 교환할 수 있다. 따라서 별도의 장비를 구입하지 않더라도 간단히 통신을 할 수 있으며, 네트워크의 구성 또한 매우 용이하다. 유닉스는 많은 네트워크를 구성하고 있는데, 그 중 대표적인 것으로 유즈네트(USENET)를 들 수 있다. 유즈네트는 유닉스를 사용하는 전세계의 컴퓨터들을 연결해 놓은 것으로, 사용자 수가 약 천만 명에 이르는 거대한 네트워크이다.

마지막으로 유닉스 환경에서 운영되는 풍부한 소프트웨어를 빼놓을 수 없는데, 유닉스 사용자가 많은 만큼 그 프로그램도 다양하여 유닉스용 데이터 베이스, 스프레드시트 및 캐드/캠용 프로그램에 이르기까지 그 종류가 다양하다.

다. Windows NT

Windows NT는 네트워크 분야로는 엔터프라이즈 와이드(Enterprise Wide, 광범위 기업형)네트워크와 클라이언트/서버 네트워크 시장을 공략하고, 운영 체제로서는 유닉스와 대응할 수 있는 즉, 워크스테이션급 컴퓨터에서 동작시킬 수 있는 32비트 운영 체제라 할 수 있을 것이다. Windows NT는 NT라는 버전과 NT AS(NT Advanced Server)라는 버전으로 나뉘어 발표 되었는데, 유닉스와 저가격으로 대응할 수 있는 기능을 강조한 것이 NT이고 네트워크 기능을 강조한 것이 바로 NT AS이다. 마이크로소프트에 의하면 Windows NT는 도스와 윈도우즈 3.1을 대체하는 완전히 새로운 운영 체제라 한다. PC사용자들은 Windows NT를 윈도우즈 3.1을 개선한 상위 버전이라고 생각하기 쉽지만, Windows NT는 윈도우즈 3.1과 완전히 다른 운영 체제이다.

▶ Windows NT의 구조

Windows NT는 기존의 운영 체제와는 완전히 다른 구조를 가지고 있다. 구조 상의 특징은 마하(Mach)커널 2.0을 사용하고 네트워크에 맞게끔 그 구조 자체가 클라이언트/서버 구조(네트워크에서의 클라이언트/서버와는 다른 개념)를 가지고 있다. 또한 사용자 부분과 커널 부분으로 이루어져 있으며, 사용자 부분에는 다른 운영 체제와의 호환성을 위한 서브 시스템이나 로그인 프로세스같은 사용자 인터페이스 부분, 그리고 윈도우즈의 새로운 API인 Win32의 서브 시스템이 위치한다. 사용자 프로세스는 커널 부분과 완전히 독립되어 있으며 사용자 프로세스가 접근할 수 있는 부분은 시스템 서비스라고 불리는 부분이다. 즉, Windows NT에서는 하드웨어에 직접 어떠한 값을 써넣거나 읽어 내는 것이 불가능하다. 커널은 NT Executive라고 부르기도 하는데, Windows NT의 특징을 가장 잘 나타내주는 부분이다. NT Executive는 시스템 서비스, 실행 컴포넌트(Executive Component), 커널, 하드웨어 추상층(Hardware Abstraction Layer)으로 나뉜다.

시스템 서비스는 말 그대로 사용자 프로세스의 요구를 NT Executive에 전달하는 역할을 한다. 쉽게 생각하면 시스템 서비스는 NT Executive의 API인 것이다. 실행 컴포넌트는 오브젝트 관리자(Object Manager), 보안 관리자(Security Reference

Monitor), 프로세스 관리자(Process Manager), 멀티 프로세스 관리자(Local Procedure Call Facility), 가상 메모리 관리자(Virtual Memory Manager), 입출력 관리자(I/O Manager)로 나뉘어져 있으며 이들이 NT Executive의 구성 요소들이다. 특히 입출력 관리자는 Windows NT의 파일 시스템, 네트워크 드라이버 등과 통신을 하는 역할을 하는데, 사용자 프로세스와 통신하는 것이 아니라, 시스템 서비스에서 받아들인 것과 통신을 한다. NT Executive에 속해 있는 커널은 NT Executive의 나머지 부분이 요구하는 운영 체제기능의 집합이다. 커널은 NT Executive의 구성 요소로서 각 쓰레드의 스케줄을 관리하고, 프로세서의 인터럽트나 예외 규정을 정의, 멀티 프로세서의 동기화, 시스템 복구 역할을 수행한다. HAL은 하드웨어를 어떻게 다루어야 하는지를 기록하고 있는 일종의 정보층(Information Layer)이다. 이 HAL을 이용하기 때문에 여러 가지 CPU에서도 동작할 수 있는 것이다.

▶ Windows NT의 특징

처음 Windows NT는 기존의 운영 체제가 가지지 못하는 많은 이점과 가능성을 제시했다. 그러나, 실제로 Windows NT가 가지고 있는 대부분의 특징은 유닉스 기반의 운영 체제에서 이미 제공하고 있는 것을 Windows NT화한 것에 불과하다고 생각할 수도 있다. 예를 들어, CPU의 제약에서 벗어난 '자유로운 이식성'부터가 유닉스에서의 기본적인 특징이었고 멀티태스킹이나 멀티프로세싱, 보안과 같은 기능들 역시 유닉스에서 이미 구현한 것들이기 때문이다. 그러나, 이런 점에도 불구하고 Windows NT는 유닉스가 가지고 있는 장점을 수용한 것 외에도 여러 가지 특징을 제공한다.

■ 네트워크 운영 체제로서의 Windows NT

Windows NT가 가지는 가장 큰 특징은 강력한 네트워크 기능이다. 이 점은 Windows NT가 네트워크 기능을 기준으로 두가지 각기 다른 버전으로 제공되는 것에서 확인할 수 있다. 일반 PC사용자를 위한 데스크탑 버전은 등배형 네트워크 기능을 통해 파일과 프린터를 공유하는 소규모 네트워크 구축에 적합한 솔루션이다. 다른 하나는 강력한 네트워크 기능을 가진 NT AS버전이다. NT AS버전의 네트워크 기능은 상당히 강력해서 고성능 네트워크와 기업 규모의 네트워크를 구성할 수 있다. 또한 보안 기능, 멀티태스킹 기능과 연계되어 더욱 강력한 기능을 제공한다. 또 하나의 특징은 뛰어난 이식성이다. 즉 여러 가지 시스템에서 동작이 가능하다. 보통 하나의 운영 체제는 하나의 CPU종류와 아키텍처에서만 동작되도록 설계되고 만들어진다. 그러나, Windows NT는 인텔 X86계열의 CPU뿐만 아니라 DEC의 알파칩과 미스의 R4000계열의 CPU에서도 동작하고, 다른 CPU에서도 동작할 수 있게 설계되었다. 이것은 네트워크 환경에서 무척 큰 의미가 있으며, 운영 체제자체로서 갖는 의미도 크다. 그리고 NT는 멀티프로세싱과 스케줄링 기능을 제공한다. Windows NT는 또한 1980년대 중반 미국 정부가 정한 유닉스 인터페이스에 기초한 인터페이스 기준인 포직스(POSIX)를 따르며, 미국 정부가 정한 보안 레벨인 C2 레벨을 준수한다.

- 새로운 동작 방식에 의한 Windows NT

일반 사용자에게 Windows NT는 자체 동작 방식에 따른 장점이 더욱 중요할 것이다. 특히, 윈도우즈 3.1에 불만을 갖고 있는 윈도우즈 사용자에게는 더욱 그렇다. Windows NT가 어떻게 동작하는가에 대한 관심도 이런 이유에서일 것이다. Windows NT의 동작 방식이란 32비트 운영 체제나 선점형 멀티태스킹, 멀티쓰레딩과 같은 말들을 가리킨다. 이러한 동작 방식의 Windows NT가 주는 최고의 장점은 응용 프로그램을 통한 생산성을 향상시킨다는 것이다. 또한 윈도우즈 3.1 사용자들에게 Windows NT가 만족감을 주는 것이 있다면 완벽한 태스크 보호이다. Windows NT의 32비트애플리케이션들은 더욱이 가상 보호 모드에서 실행되므로, 에러가 난 프로그램이 다른 프로그램에 영향을 미치지 않는다. 하나의 프로그램이 전체적인 시스템을 마비시키는 윈도우즈 3.1의 치명적인 에러에 비해 Windows NT가 주는 확실한 장점이다.

- 선점형 멀티태스킹

선점형 멀티태스킹은 강제력을 가진 멀티태스킹을 의미한다. 윈도우즈 3.1은 애플리케이션이 CPU의 제어권을 가지고 있는 방식으로 멀티태스킹을 할 수 있지만, Windows NT는 스케줄러라고 불리는 모듈이 CPU의 제어권을 어떠한 프로그램에게 넘겨줄 것인가를 결정한다. 즉 언제든지 실행중인 애플리케이션을 강제로 중단시키고 다른 프로그램에 CPU의 제어권을 넘겨줄 수 있다.

- 멀티쓰레딩

멀티쓰레딩은 한 개 또는 여러개의 애플리케이션이 호출하는 여러개의 함수를 동시에 실행시킬 수 있는 기능을 말한다. 예를 들면, 하나의 응용 프로그램을 실행시킬 경우, 하나의 함수는 프린터로 출력하는 일을 수행하고, 다른 함수는 파일을 여는 일을, 또 다른 함수는 파일을 로드하는 일을 동시에 수행할 수 있다.

- 풍부한 자원

Windows NT는 제공하는 자원(resource)이 크다. 윈도우즈 3.0의 경우는 64KB의 자원을 제공했고, 3.1은 124KB의 자원을 제공한다. 그러나 이 자원의 양은 상당히 적은 것이어서 액셀이나 MS워드 또는 여러 가지 그래픽 프로그램을 동시에 사용하는 경우는 자원 부족이라는 심각한 문제에 봉착하기도 했다. 그러나, Windows NT는 자원이 풍부하기 때문에 이러한 문제가 발생하지 않는다.

- 특유의 파일 관리 시스템(NTFS)

도스나 윈도우즈가 사용하는 파일 관리 시스템은 FAT 라는 방식인데, 이 방식은 320MB 이하의 하드디스크를 관리하는데 유효할 뿐 16비트 구조와 한계가 뚜렷한 에러 처리 등의 단점을 안고 있다. Windows NT가 사용하는 파일 시스템 방식은 NT에서 사용하기 위해 설계한 NTFS(NT File System)이다. 이 시스템은 고용량의

하드디스크를 관리하는데 적합하며, 특히 32비트 구조로 파일 이름 8자, 확장자 3자라는 말도 안되는 한계를 없애는 등 기존 FAT방식의 한계를 극복했다. 또 디스크 에러 처리에 있어서도 분명한 해결 방법을 제시하며, 모든 파일을 사용자가 정의하거나 시스템이 정의한 속성을 가진 객체로 취급함으로써 객체지향 애플리케이션을 지원한다.

라. OS/2

(447)과거 마이크로소프트와 IBM은 우리가 우호적인 관계를 가지고 있었다. 즉, 마이크로소프트는 IBM의 가장 큰 협력업체(vender) 중의 하나였다. 이 관계는 PC가 처음 나왔을 때부터 본격적으로 시작된 것으로 마이크로소프트는 IBM의 PC를 위해 DOS를 만들었으며, DOS의 환경을 개선하기 위해서 윈도우즈(Windows)를 개발했다. 이러한 와중에 IBM과 마이크로소프트는 커다란 야망을 가지게 되었다. 그것은 PC의 상위 기종인 PS/2에 장착될 좀더 진보적인 운영 체제에 대한 계획이었다. 그들은 공동으로 이 작업에 착수했다. 하지만 이러한 계획의 진행은 마이크로소프트에게 상당한 고민거리를 안겨 주었다. 그것은 만일 새로운 운영 체제가 나온다면 사용자들은 모두 그것을 쓰게 될 것이고 당연히 지금 자사의 생명이 달린 DOS 시장과 새로이 준비하고 있었던 윈도우즈에 대한 계획이 수포로 돌아가기 때문이었다. 결국 빌 게이츠(마이크로소프트)는 DOS 시장을 포기하지 않았으며 IBM과 결별하게 되었다. 한편 IBM은 마이크로소프트가 없는 상태에서 새로운 운영 체제를 더욱 발전시켜 나갔다. 여기서 말하는 새로운 운영 체제가 바로 PS/2의 운영 체제라는 것과 새로운 운영 체제라는 개념을 갖는 OS/2의(Operating System/2)이다. 실제로 OS/2와 윈도우즈 NT(혹은 윈도우즈 95)의 기능 비교에서 비슷한 기능들을 사용하고 비슷한 기술을 사용하는 것은 이러한 OS/2의 개발 배경을 보면 쉽게 이해가 간다. 윈도우즈 95의 많은 기술들이 OS/2를 개발할 당시 사용하던 기술들이다. OS/2는 초창기 PS/2에만 쓰였으며, 모든 것이 그렇겠지만 그렇게 세련된 모습을 가지고 있지 못했다. 다만 OS/2 개발의 목적 중 가장 중요한 것이 DOS가 가지고 있는 기본 메모리의 한계를 벗어나려는 것이었으므로 초창기에도 메모리 관리 면에서 상당히 우수한 모습을 보이고 있었으며, GUI(Graphical User Interface)가 관심의 대상이 되자 이 부분을 집중적으로 강화해 나갔다.

▶ OS/2의 특징⁵⁾

■ 객체지향적인 GUI 작업 환경, WPS

OS/2에서 제공하는 GUI환경은 WPS(Work Place Shell)이다. 보기에다 미려할 뿐만 아니라 요즘 관심의 대상이 되고 있는 객체지향(Object Oriented)성을 상당히 많이 띠고 있다. 기본적인 파일의 관리는 객체지향성에 바탕을 둔 오브젝트(Object)의 개념을 사용한다. 오브젝트는 쉽게 말해서 프로그램 자체(논리적) 또는 OS/2에서 이용되고 있는 모든 물리적 장치라고 생각하면 된다. 그러므로 오브젝

트로 인식되는 것은 해당 프로그램 자체이다. 이것은 파일 이름과는 다른 개념임을 알 수 있다. 오브젝트를 나타내는 아이콘과 오브젝트 이름을 마음대로 편집할 수 있으며, 원한다면 다른 것으로 바꿀 수도 있다. 또한 이들은 파일 이름과는 다르게 인식된다. 한편, 또다른 개념이 폴더(Folder)이다. 폴더는 오브젝트들이 들어가는 책상 서랍이나 서류철 같은 것이다. 폴더 속에 또다른 폴더가 올 수 있고, 오브젝트들이 올 수 있으며, 폴더의 아이콘 모양, 이름 등을 마음대로 바꿀 수 있게 되었다. 이러한 것은 객체지향성에 충분히 바탕을 두고 있기 때문이다.

WPS의 객체지향성에 대해서 잠시 설명하자면, 예를 들어 WPS에서 특정한 오브젝트를 하나의 폴더에서 다른 폴더로 이동시킨다고 가정해 보자. 예전에 사용했던 DOS의 개념을 쓰면 분명 MOVE라는 명령을 생각하게 될 것이다. 그러나, 이것은 객체지향적인 생각이 아니다. 실제로 움직여지는 것은 해당 오브젝트이다. 그런데 우리는 그것을 직접 움직이는 것이 아니고 MOVE라는 또하나의 객체를 사용해서 원하는 것을 목적지로 보낸다. 이것은 분명 비효율적인 방법이며 일상 생활과 동떨어진 것이 된다. 그러나 OS/2에서 이러한 일을 한다고 하면 간단히 마우스를 이용하여 필요한 오브젝트를 들여다가 원하는 곳에 놓기만 하면 된다. 여기서 옮겨진다는 개념은 기존 윈도우즈에서 행해지던 아이콘의 이동만을 의미하는 것이 아니라 그 프로그램 자체가 옮겨지는 것을 의미한다. 이러한 일을 드래그 앤 드롭(Drag & Drop)이라 한다. 객체지향성의 편리성은 이외에도 수 없이 많으며 그 효과 또한 대단하다. OS/2를 전혀 모르는 어린이와 컴퓨터를 이제까지 사용해온 사람이지만 OS/2를 모르는 사람과 같은 시간을 주고 OS/2를 쓰게 한다면 두 사람이 OS/2와 친숙해지는 정도는 거의 차이가 없다. 이것은 바로 OS/2가 초보자에게 쉬운 접근 방법을 제시한다는 점 때문이다. 즉, 일반적인 사용면에서 본다면 그저 응용 프로그램에 대한 지식만 가지고 있으면 된다. WPS를 사용하기 위해 배워야 할 지식은 거의 없다. 단지 마우스 다루는 법 정도만 알고 있으면 WPS를 기본적으로 사용하는 데에는 아무런 지장이 없다.

■ 강력한 파일 시스템, HPFS

HPFS를 사용하면 2가지 측면에서 하드디스크 액세스(Access)속도를 높일 수 있다. 먼저 디프래그먼트(Defragment)의 개념이다. HPFS는 파일을 저장할 때 하드디스크의 한곳에 집중적으로 저장한다. 그리고 그것이 업데이트될 때도 역시 한곳에 집중적으로 저장한다. 그러므로 하나의 프로그램을 실행할 때 하드디스크 헤드가 디스크의 이곳 저곳을 돌아다니지 않아도 된다. 이렇게 물리적으로 하드디스크 헤드가 움직이는 공간을 줄이는 것은 수행 속도를 빠르게 해준다. 이것은 디프래그먼트 개념을 씌워서 DOS에서 쓰고 있는 FAT 파일 시스템처럼 특별한 유틸리티를 이용하여 하드디스크를 정리하지 않아도 된다는 것을 의미한다.

다음은 캐시(Cache)의 이용에 의한 것이다. HPFS를 이용할 경우 OS/2는 램(RAM)을 캐시로 이용할 뿐만 아니라 하드디스크에는 내장 캐시(Built-in Cache)를 직접 제어하게 된다. 이것은 상당히 빠른 데이터 입/출력 속도를 제공한다. 물론 여기서 말하는 내장 캐시는 읽기 캐시(Reading Cache)전용으로 사용됨을 의미하

는 것이다. 그리고 OS/2에서는 메인 메모리의 모자라는 양을 하드디스크의 스왑 파일의 형태로 이용하게 되는데 대부분의 사용자들은 이러한 스왑 파일을 읽기 캐시의 개념과 혼돈하는 수가 있는데 이것은 캐시의 개념이 아니고 단지 메모리의 대용일 뿐이므로 캐시의 효율성을 떨어뜨리지 않는다. 실제로 사용자가 정의해줄 수 있는 쓰기 캐시의 경우는 메인 메모리를 이용하게 된다. 또한 읽기 캐시와 쓰기 캐시는 항상 짝을 이루고 있으며, 이들은 OS/2 자체 성능에 커다란 역할을 담당하고 있다.

HPFS는 다양한 파일 속성(Extended Attribution)을 제공한다. 예를 들면 최근의 파일이 참조된 시간이라든가 업데이트된 시간뿐만 아니라 언제 생성되었는지 어떤 형태를 가지고 있는지를 쉽게 볼 수 있게 한다. 개발자의 입장에서는 상당히 편리한 기능이다. 또한 HPFS를 이용할 경우 255자까지의 파일 이름을 쓸 수 있다. 여기서 말하는 255자는 단지 알파벳의 단순한 나열을 뜻하는 것이 아니라 문장이 될 수도 있고 제한적으로 특수 문자도 쓸 수 있게 되어 있다. 또 대소문자의 구분이 되며(실제로 사용면에서는 대소문자의 구분이 가지 않는다), FAT 파일 시스템을 동시에 사용할 경우에는 FAT 파일 시스템으로 된 파티션이라 할지라도 다양한 파일 속성을 줄 수 있게 되어 있다. 다음은 디스크 공간의 낭비 방지에 관한 것으로 일반적으로 디스크를 관리하는 개념은 트랙(Track)과 섹터(Sector)이다. 하드디스크에는 플로피디스크처럼 되어 있는 디스크(플렛)가 여러개 들어 있다. 이들 디스크 중 한 개만을 생각해 볼 때, 한 개의 디스크를 여러개의 동심원으로 나눈 것이 트랙이다. 또한 한 개의 트랙은 보통 여러개의 섹터로 나뉘어진다. 즉, 한 개의 트랙속에 여러개의 섹터가 있고, 이렇게 여러개의 섹터가 있는 트랙이 모여 수백에서 수천개 모여 하나의 디스크를 구성하며 이러한 디스크가 여러개 모여 한 개의 하드디스크를 구성하고 있다. 한편 한 개의 섹터는 512바이트로 구성된다. FAT 파일 시스템에서는 이러한 섹터가 여러개 모인(최소한 4개 이상) 클러스터(Cluster)라는 개념을 쓴다. 그래서 FAT 파일 시스템에서 데이터가 저장되는 최소 단위는 한 개의한개의 클러스터가 4개의 섹터로 구성된다고 가정하더라도 2048바이트가 된다. 하지만 HPFS는 이러한 클러스터의 개념을 쓰지 않고 직접 섹터를 이용한다. 그러니까 최소 단위가 512바이트가 되는 것이다. 그러므로 최소 단위 이하의 용량을 갖는 파일을 저장하더라도 FAT 파일 시스템에 비해 상대적으로 디스크 공간의 낭비가 적어진다.

다음은 에러 복구 능력으로 기본적으로 디프레그먼트 개념을 씀으로써 파일 관리 체인을 잃어버릴 확률을 줄였다. 그리고 만일의 경우에 대비해서 Hot Fix라는 기능을 제공한다. Hot Fix는 하드디스크 에러 복구 기능을 해주는 중요한 역할을 수행한다. 예를 들어 HPFS를 사용할 때 베드 섹터(Bad Sector)가 발생하면 자동으로 베드 섹터 내의 내용을 안정적이면서 비어 있는 공간으로 이동시킨다. 그리고 사용자에게 이제까지의 진행 상황을 알려준다. 이렇게 옮겨진 데이터는 다시 OS/2를 부팅시킬 때까지 옮겨진 곳에서 그냥 이용된다. 그리고 다시 부팅되면 자동적으로 인식된 CHeckDisk가 실행되고 옮겨졌던 데이터는 자신이 속해 있는 파일이 있던곳의 가장 가까운 공간으로 다시 옮겨지게 된다. 물론 베드 섹터가 낮

던 섹터는 다음부터 사용할 수 없도록 표시를 해준다. 따라서 사용자는 하드디스크에 관련된 문제로 시스템을 못쓰게 되는 경우를 거의 만날 수 없게 된다.

■ 멀티미디어 기능

멀티미디어란 쉽게 말해 우리가 매일 사용하고 있는 가전제품 중 미디어 기능을 하는 것들(TV, 라디오, 비디오, 전화, 오디오 등)을 컴퓨터를 이용해서 통합한다는 뜻이다. 이렇게 미디어들을 통합하기 위해서는 운영 체제 자체적인 지원이 필수적이다. 멀티미디어 시대의 흐름에 편승하여 OS/2의 모습은 멀티미디어 지원에서 훨씬 더 폭을 넓히고 있다. 근간에 한참 열기를 더해가고 있는 멀티미디어에 대한 사용자의 관심에 부응하여 기본적인 멀티미디어 솔루션을 제공하고 있는 것이다. 그 기본이 되는 부분은 역시 MPM/2(MultiMedia Presentation Manager/2)인데, 이것은 기본적으로 다양한 멀티미디어 솔루션을 제공한다. 예를 들어 Software Motion Picture는 실시간으로 특별한 장치없이 동영상을 재현할 수 있으며 또한 CD 음악을 들을 수 있는 CD플레이어를 기본으로 제공하고 있다.

마. NeXT Step

(4)(7) 애플 컴퓨터사의 창립자 스티브 잡스는 1987년 회장직에서 쫓겨난 후 애플사를 나와(NeXT)라는 새로운 회사를 설립하고, 88년 넥스트(NeXT)라는 이름의 워크스테이션급 PC를 개발했다. 넥스트는 육면체 모양을 하고 있어 큐브라는 별칭으로, 또 본체와 모니터, 키보드, 마우스까지 온통 까만 색으로 디자인 되었기 때문에 블랙 박스라고도 불리었다. 넥스트는 당시 잘 쓰이고 있지 않은 새로운 기술을 많이 도입했는데, 650MB MOD를 기본으로 장착하고 사운드 등을 처리하기 위해 DSP를 사용했다. 그러나, 이 시스템이 사람들의 주목을 끈 이유는 단순히 뛰어난 하드웨어 때문이 아니었다. 잡스는 넥스트를 만들면서 여기에 사용할 새로운 운영 체제인 넥스트 스텝(NeXT Step)을 만들었다. 이 운영 체제는 매킨토시와 같은 GUI를 제공하면서도, 멀티유저, 멀티태스킹을 지원해 사용자와 개발자 모두에게 큰 찬사를 받았다. 그러나, 당시 만불이상의 넥스트를 살 수 있는 사람은 얼마 되지 않았으며 기업이나 연구소들은 같은 가격대의 더 빠른 계산 능력을 가진 선 워크스테이션을 선호했다. 사용자들의 저가모델에 대한 요구에 따라 넥스트사는 저가의 NeXTstation을 내놓았고 컬러에 대한 사용자들의 요구에 대해 트루 칼라를 지원하는 그래픽 보드를 판매했다. 그러나, 이미 IBM호환PC와 선호환 스테이션이 대세인 시장에서 독자적인 하드웨어를 판다는 것은 매우 힘든 일이었다. 넥스트는 매우 좋은 기계임에도 사기에는 망설여지는 기계였고, 넥스트 스텝은 사용하고 싶어도 사용할 수 없는 운영 체제가 될 수 밖에 없었다. 이런 상황에서 넥스트사는 결국 독자적인 하드웨어를 포기하고 시스템 소프트웨어 전문회사로 방향을 바꾸었다. 넥스트 스텝을 차세대 운영 체제로서 PC를 비롯한 모든 하드웨어에서 사용하게 만들겠다는 것이다. 그 첫 제품이 넥스트 스텝 for 인텔 프로세서(이하 넥스트 스텝 FIP)이다. 넥스트 스텝 FIP는 인텔 80486DX 이상의 CPU를 가진 기계에서 동작한다. 즉 486DX나 486SX+487SX 또는 펜티엄 시스템에서 동

작한다. 물론 이 외에도 HP의 PA-RISC와 선의 워크스테이션에도 이식 작업이 진행 중이다.

▶ **넥스트 스텝의 특징**

넥스트 스텝은 유닉스를 기반으로한 멀티유저, 멀티태스킹 운영 체제이다. 완전히 새로운 운영 체제가 아니라 기존의 BSD 유닉스에 GUI를 추가한 것이다. 그리고 OS/2나 윈도우즈와 같이 PC용으로 개발된 운영 체제가 아니고, 원래 워크스테이션급 시스템에 맞춰 개발되었기 때문에 PC에서 쓰기에는 덩치가 큰 편이다. 넥스트 스텝은 크게 커널 부분과 GUI부분으로 나눌 수 있는데, 최하층 구조인 커널은 BSD 유닉스로 기존 유닉스의 기능을 모두 지원한다. 그러나, 내부적으로는 순수한 BSD 유닉스가 아니라 카네기 멜론 대학의 마하 마이크로 커널(Mach Micro Kenerl)을 기반으로 하기 때문에 멀티 프로세서, 멀티 쓰레드(multi thread),메세지를 기반으로한 IPC(Inter Process Communication)등 32비트 운영 체제들이 내세우는 기능 대부분을 지원한다. 특히 마하 커널은 네트워크 환경과 클라이언트 서버 구조에 유리하다. 그러나, 넥스트 스텝과 다른 운영 체제와의 가장 큰 차이는 GUI부분에 있다. 넥스트 스텝은 디스플레이 포스트스크립트(Display Post Script)와 객체지향 API를 자랑한다.

■ **디스플레이 포스트스크립트**

넥스트 스텝은 디스플레이 포스트스크립트를 기반으로한 윈도우 서버를 제공한다. 디스플레이 포스트스크립트는 현재 많은 프린터에서 사용하는 포스트스크립트를 화면 출력에 적합하게 확장한 것으로 포스트스크립트의 모든 기능을 지원한다. 이를 통해 강력한 그래픽 기능을 손쉽게 사용할 수 있는데, 포스트스크립트를 이용함으로써 사용자는 단순히 기본적으로 제공되는 그래픽 함수만 사용하는 것이 아니라, 자신의 포스트스크립트 함수를 다운로드해 기능을 확장할 수 있다. 또한 시스템에 새로운 기능을 추가하거나 확장하기도 매우 용이하다. 디스플레이 포스트스크립트의 또다른 장점은 화면 출력과 프린트 출력을 위해 프로그램을 따로 만들 필요가 없다는 것이다. 또한 고급 그래픽 소프트웨어인 경우, 포스트스크립트의 기능을 활용하기 위해서 포스트스크립트를 위한 출력 루틴을 가지고 있는 것이 대부분이다. 이런 점에서 디스플레이 포스트스크립트는 그래픽 처리가 많은 프로그래밍에 큰 도움을 준다.

■ **객체지향 API**

넥스트 스텝의 가장 큰 특징은 무엇보다도 객체지향적인 개발 환경이다. 만일 프로그래머가 넥스트 스텝용 애플리케이션을 개발하려면 기본적으로 오브젝티브 C(Objective-C)라는 객체지향 언어를 써야 하며, 넥스트 스텝의 API는 Objective-C 클래스 라이브러리로 구성되어 있다. 넥스트 스텝에서는 각 API를 종류별로 묶어 각각을 하나의 키트(kit)로 호칭한다. 이 키트 중에서 가장 기본이 되는 것이 모든 애플리케이션의 기본 구조를 결정하는 AppKit(Application Kit)이다. AppiKit에는 애플

플리케이션, 메뉴, 버튼, 윈도우 등 프로그램과 사용자 인터페이스의 기본 요소들을 모두 클래스로 제공하며, 프로그래머는 이들 기본 클래스들을 짜 맞추거나 서브클래스로 확장함으로써 손쉽게 자신의 프로그램을 짤 수 있다. AppKit외에도 SoundKit, MusicKit,DBKit 등이 제공되며 최근에는 3차원 렌더링을 지원하는 3DKit가 추가됐다.

■ 강력한 네트워크

넥스트 스텝은 다른 유닉스 계열의 운영 체제와 마찬가지로 TCP/IP를 기반으로 하는 네트워크와 NFS를 지원한다. 그러나, 넥스트 스텝은 이 외에도 PC에서 많이 쓰이는 NOS인 노벨의 네트웨어와 매킨토시용 네트워크인 이더토크(Ether Talk)를 지원한다. 즉 이미 네트워크가 설치된 PC 사무환경에서 넥스트 스텝을 그대로 사용할 수 있으며, 유닉스의 FTP, 텔넷, E-MAIL 등을 함께 사용할 수 있다. 또한 넥스트 스텝은 윈도우즈의 OLE에 해당하는 오브젝트 링킹(Object Linking)을 단순히 한 시스템 안에서 뿐만 아니라 분산 환경에서도 사용할 수 있다. 넥스트 스텝을 사용하는 LAN상의 다른 시스템과도 객체연결이 가능하다는 것이다.

■ GUI를 위한 개발 환경도 GUI

넥스트 스텝 역시 편리한 개발 환경을 자랑한다. 일반적으로 GUI 환경은 사용자에게는 편리하지만 개발자들은 이를 지원하기 위해 텍스트 환경에 비해 훨씬 많은 일을 해야 한다. 넥스트사는 넥스트 스텝에서 애플리케이션을 개발하는 것이 다른 GUI 환경에 비해 5배 정도 빠르고 효율적이란 사실은 누구나 인정하고 있다. 넥스트 스텝의 개발 효율은 무엇보다도 앞에 언급한 객체지향적인 API와 이를 지원하는 비주요한 개발 툴에서 비롯된다. GUI는 기본적으로 객체지향적일수록 편리하다. 넥스트 스텝은 프로그램 작성을 위한 기본 언어로 오브젝트-C를 사용한다.

바. Linux

최초의 리눅스는 핀란드의 Linus Torvalds라는 학생이 혼자 힘으로 개발하였다. 그는 또다른 유닉스 클론인 미닉스(Minix)를 보고 리눅스를 만들 생각을 하게 되었다. 그러나, 처음에는 혼자서 모든 코딩을 다 했으나 곧 이소식을 들은 GNU의 협력을 받게 되었으며, 전세계 많은 프로그래머들의 참여와 도움을 받게 되었다. 최초의 리눅스는 91년도 0.01 버전에서 시작했으나 이 버전은 사실 등록조차 제대로 되지 않은 것이었다. 그러나 비교적 짧은 시간 내에 많은 변화를 거듭하여 1994년 3월14일, 드디어 버전 1.0이 발표되었다. 이런발전이 있기까지는 인터넷이라는 국제적인 통신망이 지대한 역할을 했음은 물론이다.

▶ 리눅스의 특징

다른 32비트 운영 체제들과 리눅스의 가장 큰 차이를 들라면, 리눅스는 상용이

아닌 완전한 공개 소프트웨어라는 점이다. 리눅스는 누구나 자유롭게 복사해서 사용할 수 있으며, 심지어 소스 코드 자체도 공개되어 있다. 리눅스에 포함된 대부분의 소프트웨어는 'copy left(상용은 copy right다)'라는 GNU의 저작권 규칙에 의해 취급된다. 리눅스는 일반 유닉스 계열로 볼 수 있는 다음과 같은 특징을 갖고 있다. 우선 완벽한 멀티태스킹, 멀티유저 환경이 지원된다. 또 프로세스간의 완벽한 보호 기능(프로텍션)을 제공하며, 실행 파일의 요구 적재(demand load, 프로그램 실행시 파일 전체를 메모리에 읽어들이는 것이 아니라 실제 사용되는 부분만을 읽어 들이는 기법)라든가, 쓰기시 복사(copy-on-write, 한 프로그램을 동시에 여러 번 실행할 경우 일단 각 프로세스가 같은 메모리를 공유하다가, 한 프로세스에서 쓰기를 할 경우에 한해 그 프로세스에 메모리를 할당하는 방법) 등의 고급기법을 통해 메모리 이용 효율을 극대화하고 있다. 리눅스에서는 페이징 기법을 사용해 메모리 관리를 하며, 동적 링크 라이브러리(Dynamic Link Library, DLL)를 제공한다. 또 최대 256MB까지 스왑핑을 허용하므로 큰 프로그램도 수행 가능하다. 호환성 측면에서는 포직스, 시스템 V, BSD 등과 소스 레벨에서 호환성을 제공한다. 또 csh 등에서 사용되는 포직스 작업 제어방식을 지원하는데, 여러 작업을 백그라운드로 수행하며, 작업간 전환을 간편하게 지원한다. 또 완벽한 TCP/IP 네트워킹을 제공한다. 그외에 리눅스만의 특징을 들자면, 여러 개의 가상 터미널을 지원하기 때문에 단축키를 누르는 것만으로도 한 대의 PC로 마치 여러대의 터미널을 쓰는 것과 같은 효과를 낼 수 있다. 또한 운영 체제 차원에서 387코프로세서 에뮬레이션을 제공하기 때문에 응용 프로그램에서 에뮬레이션을 할 필요가 없고, 다양한 국가별 키보드 형식을 지원한다. 지원하는 파일 시스템도 다양하다. 기존의 미닉스, 제닉스, 시스템 V 계열의 파일 시스템을 지원한다. 또 도스용 파티션 리눅스 디렉토리 구조에 마운트하여 자유롭게 액세스가 가능하며, OS/2의 HPFS의 경우는 읽기만 가능하다. 그외 최근 많이 보급되고 있는 CD롬의 파일 시스템인 ISO-9660도 지원한다.

■ 프로그램 개발 환경

리눅스에서는 컴파일러, 라이브러리, 디버거 등의 프로그래밍 환경이 완벽하게 지원된다. 유닉스 계열에서 프로그래밍은 C/C++가 기본인데, 리눅스에서는 기본적으로 GNU의 C/C++ 컴파일러인 gcc를 지원한다. gcc는 ANCI C를 완벽히 지원하며, 이식성이 뛰어나고 효율이 좋은 코드를 만들어 내기 때문에 상용에 비해 전혀 손색이 없다. 이외에 스포토크, 파스칼, 포트란, Lisp 등의 언어도 지원한다. 또 디버거인 gdb, 대형 응용 프로그램 개발에 유용한 make, imake 등이 있으며, bison, flex는 기존 yacc, lex의 기능을 대체한다.

■ X 윈도우

X 윈도우는 여러 유닉스 계열 운영 체제에서 표준으로 사용하는 윈도우 기반의 그래픽 사용자 인터페이스이다. 원래는 MIT에서 내부적으로 쓸 목적으로 개발되었으나, 이것이 널리 퍼져 표준의 위치를 점하게 되었으며, 상용으로 나오는 GUI

들도 대부분 X 윈도우에 기반한 것이 많다. 참고로 흔히 X 윈도우가 PC의 MS 윈도우와 대등한 역할을 하는 것으로 오해하는 경우가 많은데, 물론 사용자 인터페이스라는 측면에서는 기능이 유사하다. 그러나, 윈도우즈가 운영 체제의 역할을 하는 반면, X 윈도우의 경우는 단순히 사용자 인터페이스만을 담당한다는 점이 다르다. 또 X 윈도우는 여러 대의 워크스테이션이 네트워크로 연결되어 있는 분산 환경을 가정하여 만들기 때문에 네트워킹 기능이 매우 우수하며, 한 시스템에서 수행되는 프로그램의 윈도우를 다른 시스템의 화면에 출력시켜 작업할 수도 있다. 한편 MS 윈도우즈의 경우, 윈도우 모양이 고정되어 있지만 X 윈도우에서는 다양한 윈도우 매니저를 선택해 쓸 수 있다.

■ 네트워킹

리눅스는 기본으로 TCP/IP에 기반한 네트워킹 환경을 제공한다. 따라서 이더넷 등으로 LAN이 연결되어 있다면, 전자우편(E-Mail),TELENET,ftp 등의 기본적인 기능과 NFS(Network File System), NIS(Network Information System) 등의 고급기능을 모두 사용할 수 있다. 인터넷에 연결되어 있는 경우라면 뉴스그룹 등의 이용도 가능하다. SLIP, PLIP 등도 지원하는데, 이는 각각 직렬 포트 및 병렬 포트를 통해 IP 프로토콜을 사용하기 위한 규격이다. 또 우리가 PC 통신 등에서 많이 사용하는 커밋(Kermit)이나 szrz(Z 모뎀)등이 포함되어 있고,통신용 프로그램도 제공되므로 간단하게 모뎀을 통한 통신을 활용할 수 있다.

4. 결 론

위와 같이 운영 체제에 대해 알아 보았다. 이제 앞으로 나올 운영 체제에 대해 알아 보겠다. 위의 내용들을 종합해 볼 때 우선 공통된 사항은

첫째, GUI, 멀티태스킹, 멀티쓰레딩과 같은 사용자 환경의 개선이다.

둘째, PC의 성능 향상으로 워크스테이션급 운영 체제도 모두 PC의 운영 체제로 되었다는 점이다

셋째, 그래픽 기능의 강화이다.

넷째, 멀티미디어 기능의 강화로 사용자에게 오락을 줄 수 있는 요소가 많아졌다.

다섯째, 파일 관리 기능의 강화이다.

여섯째, DOS와 몇 가지 운영 체제를 제외한 나머지 운영 체제들은 32비트 방식을 채택했다.

일곱째, 네트워킹과 통신(특히 인터넷)기능의 강화이다.

위의 사항은 현재 만들어지고 있는 차세대 운영 체제에 모두 포함되는 항목이다. 최근에는 멀티미디어와 인터넷과 네트워크가 PC활용의 필수과목으로 되어 있어 더 많은 요구가 운영 체제에 요구된다. 고화질, 고음질의 멀티미디어 데이터를 얼마만큼 지원하는가,인터넷은 편한가,호스트와의 접속을 비롯한 네트워킹이 쉬운가 등이 추가된 과목의 핵심이다.

▶ OS들의 전쟁

현재 PC OS의 가장 큰줄기는 마이크로소프트와 IBM이다. 양사는 간판 OS인 윈도우즈와 OS/2의 경쟁관계는 펜티엄 PC에서도 거듭되고 있다. 하지만 과거 PC OS 시장이 윈도우즈와 OS/2 만의 대립관계를 보였던 것과는 달리 펜티엄 PC에서는 윈도우즈 NT 워크스테이션 버전과 OS/2 멀린, 그리고 리눅스를 비롯한 PC용 유닉스들까지 가세해 다소 혼란스러운 구도가 그려질 전망이다.

많은 사람들은 윈도우즈 차세대 운영 체제로 윈도우즈 95나 97대신 윈도우즈 NT를 꼽는 것은 마이크로소프트의 전략이 윈도우즈 NT에 집중되어 있기 때문이다. 또 마이크로소프트가 1996년 7월31일 발표한 윈도우즈 NT 4.0 워크스테이션 버전을 보면 현재 일반화된 펜티엄 PC에서도 제성능을 충분히 발휘할 수 있다. 외형상의 인터페이스도 윈도우즈 95와 닮았고, 윈도우즈 95의 애플리케이션도 윈도우즈 NT 4.0에서 실행된다. 때문에 마이크로소프트가 당초 예상한 것보다 빠른 속도로 하드웨어가 발전하고 있어 윈도우즈 97보다는 윈도우즈 NT가 윈도우즈 95를 대신할 것이라는 주장이 설득력있게 들린다.

OS/2쪽에서는 멀린이 OS/2 워프의 뒤를 이을 것이라는데 이의를 제기하는 사람은 없다. OS/2 멀린은 실행시키기 위한 사양으로 보나 멀린이 가진 기능으로 보나 마이크로소프트의 윈도우즈 NT 4.0 워크스테이션 버전의 강력한 라이벌로 꼽을 만하다.

유닉스 진영에서는 이미 나와 있는 PC용 유닉스 중 가장 많은 사용자를 확보한 리눅스가 유닉스를 대표해서 경쟁에 참여할 것으로 보인다. 리눅스는 윈도우즈 NT와 마찬가지로 인텔CPU뿐만 아니라 선사의 스팩 스테이션이나 DEC사의 알파칩에도 이식할 수 있어 이식성이 있어 사용자층이 높은 편이다. 또 윈도우즈 패밀리와 함께 쓸 수 있는데다 상용이 아니기 때문에 윈도우즈 NT와 OS/2 멀린의 시장 자체를 잠식하는 것은 아니지만 PC용 운영 체제시장에서 변수로 작용할 전망이다. 다음 운영 체제시장에 변수로 떠올르는 것이 바로 넷스케이프이다. 현실적으로 운영 체제들이 가장 신경을 곤두세우는 분야는 인터넷이다. 인터넷이 PC 활용의 주목적이 되는 것을 가정한다면, PC에 가장 필요한 소프트웨어는 운영 체제가 아니라 웹브라우저이다. 넷스케이프는 윈도우즈든 OS/2든 리눅스든 운영 체제를 가리지 않는 웹브라우저이기 때문에 넷스케이프를 쓰는 사람들은 운영 체제가 무엇인지를 가릴 필요가 없다. 만약 넷스케이프가 웹브라우저의 표준이 되고 운영 체제의 기능까지 갖추었다면 다른 운영 체제들의 영역이 크게 줄어들 것이다.

결론적으로 운영 체제는 하드웨어 발달에 따른 64비트 체제로 발전할 것이지만 앞으로 인터넷의 비중이 커짐으로서 웹브라우저자체가 운영 체제로 될 것이라고 주장하는 바이다. *

* 참고 문헌

- (1) 마이크로 소프트웨어 94년 8월호
- (2) Peter Norton's DOS 5.0 Guide p.2-14

- (3) 마이크로 소프트웨어 95년 8월호
- (4) 헬로우 PC 96년 9월호 p.289-319
- (5) 윈도우 세계 96년 8월호 p.244-274
- (6) 윈도우 세계 96년 10월호 p.49-75
- (7) 마이크로 소프트웨어 95년 12월호 p.500-515
- (8) 마이크로 소프트웨어 95년 10월호 p.242-208



