

From NT to Linux

An application migration guide(2004. 03)

NT에서
Linux로의
마이그레이션
가이드

IBM[®]



From NT to Linux



NT에서 Linux로의 마이그레이션 가이드

목차

1
2
3
4

1. 왜 Linux로 마이그레이션해야 하는가 4

1.1 Linux의 사용상 장점	6
유연성	6
안정성	6
보안성	7
확장성	7
하드웨어 최적화로 부담 없는 리소스 요구 사항	7
가격	8
폭넓은 기술 지원 및 서비스 가능	8
급속도로 성장하는 시장	8
개발 리소스로 접근 가능	8
1.2 왜 Linux가 NT를 대체하는 OS인가	9

*from
NT
to
Linux*

2. 애플리케이션 마이그레이션 전략 및 고려 사항 10

2.1 마이그레이션 이전에 고려할 문제들	12
하드웨어 환경	12
소프트웨어 환경	12
기술 준비 사항	12
2.2 계획 및 준비	13
마이그레이션 계획	13
준비 작업	13
2.3 포팅	14
가능한 프로그래밍 모델 마이그레이션	14
주의해야 하는 사항	15
포팅 과정	18
포팅 툴	18
2.4 검사	19
2.5 성능 튜닝	20
2.6 구축	21

02
+
03

3. 미들웨어 마이그레이션 22

3.1 MS Exchange 서버에서 Domino 마이그레이션	24
사전 평가	24
포팅	25
검정 및 검사	26
성능 튜닝	28
레퍼런스	28
3.2 SQL 서버에서 DB2 마이그레이션	30
사전 평가	30
포팅	31
검정 및 검사	32
성능 튜닝	33

4. NT 서비스 마이그레이션 36

4.1 마이그레이션 고려 사항	38
4.2 유저 인증	39
4.3 파일 및 프린트 공유 서비스	40
4.4 그 밖의 기본 네트워크 서비스	41

1

-
- 1
 - 2
 - 3
 - 4

*from
NT
to
Linux*

Why move to Linux?

왜 Linux로
마이그레이션
해야 하는가?

다양한 애플리케이션을 위한 매력적인 운영체제인 Linux는 다양한 비즈니스 이점을 제공한다. Window NT에서 Linux 마이그레이션에 관한 세부적인 논의에 들어가기 앞서, Linux 운영체제의 여러 장점을 소개하고자 한다.

1.1

Linux의 사용상 장점

- 1
- 2
- 3
- 4

유연성

Linux는 소스가 공개된 운영체제이다. Linux 커널 및 Linux 배포판에 포함된 대부분의 소프트웨어 구성 요소들은 GPL 혹은 LGPL에 의해 라이선스를 받는다.

모든 프로그래머들은 GPL/LGPL 조항에 동의할 경우 Linux 소스 코드를 보고 변경할 수 있다. 이러한 자유성은 더욱 쉽고 빠른 문제 해결로 이어진다. 소프트웨어 버그가 발견되었을 때, 업체가 제공하는 패치 프로그램을 오랫동안 기다릴 필요 없이 사용자 혹은 커뮤니티의 다른 개발자와의 대화를 통한 오류 부분에 대한 즉각적인 해결이 가능하다. 이 사실은 단순히 이론상의 장점만을 이야기하는 것은 아니다. SAP와 같은 많은 ISV 업체들은 자사의 제품을 Linux 기반에서 안정적 사용할 수 있도록 하기 위하여 소스 코드를 최적화하였다. 따라서, 솔루션 업체들이 소스 코드를 리눅스 기반에서 안정적으로 구현하도록 수정할 수 있는 유연성 및 개방성이 Linux가 확산되는 가장 중요한 이유 중의 하나로 생각하고 있다.

설령 사용자가 Linux 기반의 공개 소프트웨어 모델에서 소스 코드를 자유롭게 수정할 수 있는 기능을 사용하지 못한다 할지라도, 유연성에 관련된 여러 다른 혜택들을 얻을 수 있다. Linux는 PDA를 비롯하여, 워크스테이션에서부터 IBM zSeries에 이르는 거의 모든 CPU 아키텍처(x86, PowerPC, SPARC, Alpha, PA-RISC, Itanium, Arm 등)를 지원한다. Linux는 TCP/IP 스택, IP 라우팅 및 Firewall 기능과 IPv6가 Linux 커널에 절묘하게 통합된 아주 강력한 네트워크 운영체제이다. Linux는 또한 Ext3, ReiserFS, JFS, BFS, NFS, FAT32, NTFS 등을 포함하는 수많은 파일 시스템을 지원하며, 이러한 지원 기능은 Linux를 다른 플랫폼에 비해 경쟁 우위에 올려 놓는다.

안정성

Linux는 Linux에 포함된 여러 훌륭한 기능들로 격찬을 받아왔다. Gartner Group이 발표한 OS 평가 리포트에 의하면, Linux는 안정성 면에서 Windows NT 및 Window 2000을 훨씬 능가하는 RISC UNIX와 함께 안정성 부분에서 "Excellent" 어워드를 수상하였다.

*from
NT
to
Linux*

보안성

Linux 개방형 표준의 특징은 폐쇄적 운영체제보다 더욱 훌륭한 보안 기능을 제공한다. Linux 코드는 수천 명의 개발자에 의해서 수정 및 보완이 되었으며, 심지어 개발 도중에도 문제점 및 버그의 발견 및 조기 수정이 가능하다. 또한, Unix와 유사한 OS인 Linux는 외부 공격에 큰 내성을 지니는 기능을 탑재하고 있다. 그 예로 “chroot”을 들 수 있다.

확장성

이전 Linux는 SMP(symmetric multi-processor) 지원에 있어 일반 기업체에서 발표한 UNIX 운영체제에 뒤쳐진 OS였다. 그러나 커널 2.4의 도입과 함께, Linux는 4-Way SMP를 강력히 지원하며 8-Way 웹 서버 지원에도 안정적이면서 뛰어난 성능을 보이고 있다. Linux는 클러스터링 기술과 함께 사용되면서 강력한 연산 작업을 통한 많은 워크로드를 처리할 수 있게 되었다. Shell의 지구 물리학 분석을 위한 1024 노드 클러스터 시스템과 8000 노드인 Google의 Linux 검색 엔진은 Linux 클러스터링의 좋은 사용 사례이다.

하드웨어 최적화로 부담 없는 리소스 요구 사항

Linux 소프트웨어는 일반 운영체제보다 크기가 더욱 작고 속도는 더욱 빠르다. Linux는 대용량의 메모리 및 빠른 CPU를 잘 이용할 수도 있지만, 이들 없이도 Linux 애플리케이션은 잘 실행이 되고, Intel 486과 같은 CPU에서도 실행된다. 또한, 서버에서 사용 시에 Linux는 GUI 및 X 서버를 사용하지 않는 편리한 구성이 가능하다. 그러므로, 많은 시스템 리소스가 절약되고, 성능이 해당 하드웨어에서 최적화된다.

가격

“공개 소프트웨어”는 그 어떤 비용도 부과되지 않음을 의미하지는 않지만, Linux를 사용함으로써 예산에는 여유가 생길 것이다. Linux는 저렴한 가격에 구입할 수 있으며 시트 비용(Seat Fee)이 부과되지 않는다. 추가 비용 없이 다양한 애플리케이션 및 툴 패키지를 CD 배포판과 함께 제공받을 수 있으며, 개발자 커뮤니티에서 제공되는 공개 패키지를 꾸준히 다운받을 수 있다. 이러한 공개 소프트웨어를 무상으로 사용할 수 있는 것이 Linux 사용 시의 비용적인 가장 큰 매력이다.

*from
NT
to
Linux*

폭넓은 기술 지원 및 서비스 가능

많은 업체들은 관리/개발 컨설팅, 강의, 프로그래밍 훈련 등을 포함하는 Linux 지원 서비스를 제공하고 있다. 이러한 업체들은 단지 Redhat, SuSe, TurboLinux 및 그 밖의 Linux 배포 업체뿐만 아니라 IBM, HP, Dell 및 세계적 혹은 해당 국가의 Linux 단체와 같은 다양한 글로벌 시스템 업체들을 포함한다. 다양한 Linux 기술 관련 문서들을 인터넷 상에서 입수할 수 있다. 또한, 수천 명의 Linux 기술 보유자들은 여러 뉴스 그룹 및 메일 리스트를 통하여 각 사용자의 연락을 기다리고 있다.

급속도로 성장하는 시장

Linux는 가장 빠르게 성장하는 서버 운영체제이다. 이러한 사실은 리서치 전문기관인 IDC, Gartner 및 여러 IT 컨설팅 업체에 의해 계속적으로 보고되고 있다.

개발 리소스로 접근 가능

Linux 프로그래머들은 Low-End Linux 플랫폼상에서 애플리케이션을 개발하여 High-End 대용량 서버로 이전하여 사용할 수 있다. 게다가, OSDL 및 IBM은 수많은 Open Source 프로젝트를 후원하고 있고 이로 인해 수많은 프로그래머들은 High-End 상용 서버 환경하에서 리눅스를 구현하고 있다.

1.2

왜 Linux가

NT를 대체하는 OS인가

08
+
09

Microsoft는 2004년 12월 31일부로 Window NT 지원을 공식적으로 중단할 것이다. 마이그레이션은 모든 Window NT 유저들에게 심각하게 고려해야만 하는 사항이 된 셈이다

이 가이드 도입 부분에서 많은 Linux 특징점에 대하여 설명하였다. 그러나, 이것은 아무 것도 없는 백지 상태에서 애플리케이션을 구축할 때 Linux가 최상의 선택임을 입증하는 것이다. 마이그레이션에 대하여 논의할 때, 마이그레이션에 관한 결정을 내리는 고객의 관점에서 비용을 고려해야만 한다. 물론, 고객은 당연히 단기간 및 장기간에 걸친 상황에 대한 비용을 고려할 것이다.

아래의 내용은 Window NT 최고의 대안인 Linux를 지지하는 IBM의 의견이다.

1. Linux는 Window 운영체제보다 더욱 안정적인 OS이다.

보안성은 비즈니스의 종류에 상관없이 고려해야 하는 중요한 문제임에 틀림없다. 악의적인 공격에 의해 시스템이 충돌하면, 비즈니스가 방해받는 것뿐만 아니라 크리티컬한 비즈니스 데이터 역시 도난당하거나 파괴된다. 결과는 회사에 치명적일 것이며, 공공기관인 경우 국가의 이익이 위협에 처하게 된다. 첫 번째 장에서 언급되었듯이, Linux는 독점적 OS보다 기본적으로 더욱 안정적이다. 또한, 보안 취약점이 드러났을 때도 빠르고 잦은 패치 제공은 시스템이 더욱 내성을 갖추도록 만든다.

2. 독점적 솔루션에 비해 더욱 저렴한 Linux 기반 솔루션은 총소유비용(Total Cost of Ownership)을 낮춘다.

- Linux는 사유 솔루션에 비해 저렴한 라이선스 비용을 요구한다.
 - 훌륭한 안정성 및 보안성은 비즈니스 비용을 절약한다.
 - 유비쿼터스 지원 플랫폼은 기존 하드웨어를 더욱 잘 활용할 수 있게 해주며 새로운 투자 비용을 줄인다.
 - Unix 기술의 사용은 애플리케이션 개발 및 시스템 유지/보수 비용을 줄인다.
 - 개방형 플랫폼은 업체의 이탈방지(Lock-in) 전략을 피할 수 있으며, 그 결과 장기간의 비즈니스 비용을 줄일 수 있다.
-

2

1

• 2

3

4

*from
NT
to
Linux*

Application Migration strategy and considerations?

애플리케이션
마이그레이션 전략
및 고려 사항

2.1

마이그레이션 이전에 고려할 문제들

1

• 2

3

4

애플리케이션 코드의 마이그레이션 전에, 몇 가지 고려해야 할 사항이 있다.

하드웨어 환경

- 어떠한 하드웨어 플랫폼으로 마이그레이션하고자 하는가? Linux는 Alpha, x86, Itanium 64, IBM zSeries/pSeries/iSeries와 같은 수많은 CPU 아키텍처에서 지원된다.
- 애플리케이션 마이그레이션이 32비트에서 64비트로의 이동인가 아니면 그 반대인가?
- 현재 하드웨어 환경이 충분한가? 고성능 네트워크 프린터 혹은 고속 Fiber Channel 디스크 어레이와 같은 새로운 장치들을 추가하고 싶은가?

소프트웨어 환경

- 사용자의 애플리케이션 개발 및 실행에 필요한 모든 개발 툴, 라이브러리 및 미들웨어가 대상 Linux 플랫폼에 사용 가능한가?
- 사용자의 애플리케이션 아키텍처상에서 구성 요소들의 변동 사항이 있는가? (예, 데이터베이스 시스템이 MS SQL에서 DB2로 변경 예정)
- RedHat, SuSe, TurboLinux 등과 같이 다양한 Linux 배포 업체들 중에서, 어느 배포판을 선택할 것인가?
- 사용자가 선택한 Linux 배포판이 사용자의 애플리케이션에 필요한 모든 하드웨어 장치를 지원할 수 있는가? 예를 들어, 선택한 Linux 배포판 커널이 귀사의 이더넷 네트워크 인터페이스, 내부 RAID 카드 및 화상 캡처 장치를 지원할 수 있는 최신의 드라이버를 갖추고 있는가?
- 사용자의 싱글 애플리케이션 소스가 하나 이상의 플랫폼을 지원할 계획인가?
- C/S 혹은 n-Tier 모델과 같은 애플리케이션 아키텍처 모델의 목표는 무엇인가? 그리고 아키텍처상의 어떠한 요소가 우선 마이그레이션 대상인가?

기술 준비 사항

- 현 개발자들이 애플리케이션 포팅을 위하여 어떤 기술 교육을 받아야 하는가?
- 현 시스템 운영자들이 마이그레이션될 시스템을 위하여 필요한 기술은 무엇인가?
- 현 애플리케이션 유저들이 마이그레이션될 애플리케이션을 위하여 필요로 한 교육이 무엇인가?

from
NT
to
Linux

2.2

계획 및 준비

12
+
13

마이그레이션 계획

마이그레이션 계획은 마이그레이션 사전 평가 후에 준비되어야 한다. 마이그레이션 계획은 마이그레이션 프로젝트의 중요 관점에 관한 조정 및 스케줄을 잡는 것을 의미한다. 보통, 회사의 마이그레이션 계획은 아래의 사항을 포함하여야 한다.

- **마이그레이션 예산** — 각각의 프로젝트에 얼마의 비용이 필요한가?
- **마이그레이션 진행 스케줄** — 마이그레이션의 각 단계에서 어떠한 목표가 달성되어야 하는가? 프로젝트는 어떻게 나누어지는가?
- **설비 리스트** — 마이그레이션 관련 모든 하드웨어 및 소프트웨어 리스트 작성
- **예상치 못한 상황 대처** — 마이그레이션 도중 예상치 못한 상황의 발생이 가능하다. 이 역시 사전 고려 대상이다.

준비 작업

- **하드웨어 환경** — 마이그레이션 준비에 필요한 모든 하드웨어 장비가 준비되어야 한다.
 - **소프트웨어 환경** — 마이그레이션 준비에 필요한 모든 소프트웨어(개발 툴, 버전 컨트롤 툴 등)가 준비되어야 한다.
 - **인적 자원** — 개발자들 교육을 통하여 애플리케이션 포팅에 준비할 수 있도록 대비하라.
-

2.3

포팅

1

• 2

3

4

가능한 프로그래밍 모델 마이그레이션

포팅 과정은 각각의 단일 애플리케이션 관련 프로그래밍 모델과 아주 밀접한 관계에 있다. 보통, 애플리케이션 아키텍처상의 상이한 부분들은 다른 프로그래밍 언어 혹은 API를 사용하면서 구축된다. 아래의 내용은 Window NT에서 Linux로 포팅 시, 각각의 애플리케이션 구성 요소와 관련된 프로그래밍 모델의 변화를 기술하고 있다.

*from
NT
to
Linux*

1. C/S 아키텍처상의 특정 클라이언트 측 소프트웨어. 이것은 보통 Visual C++ 혹은 Window Workstation상의 Visual Basic을 사용하여 프로그래밍되며 GUI(Graphic User Interface) 환경에서 실행된다. Linux에서 C/C++는 소프트웨어를 코드화하기 위하여 사용되며 GUI는 GTK 혹은 QT 라이브러리가 될 수 있다. GTK가 LGPL에 라이선스를 받은 반면, QT는 GPL의 라이선스를 받았다. 그리고 GTK는 C 라이브러리를 통하여 제공되는 반면, QT는 C++ 클래스 라이브러리를 통하여 제공된다. 이 두 API는 그래픽 기능을 완성할 수 있다. 그 결과, 코드가 사적으로 보호되어야만 한다면 GTK가 추천된다. 그리고 GTK 장치를 캡슐화하기 위하여 C++ 클래스로 정의할 수 있다. 이와는 별개로, 너무나 많은 GUI 작업이 들어왔을 때, 코드를 대규모로 새롭게 쓰는 방법이 요구된다. 비용을 고려하자면, 이러한 포팅은 마이그레이션 전체에는 필요치 않을 수도 있다. Windows Workstation을 계속 사용할 수 있고 만들어지는 애플리케이션 구성 요소(예: 임베디드 SQL)가 요구하는 변경 사항들만을 계속해서 사용할 수도 있다.
2. C/S 아키텍처의 서버 측 소프트웨어. 이것은 Window NT 서버상의 Visual C++ 혹은 Visual Basic을 사용하여 프로그래밍되며 GUI는 포함하지 않는다. Linux상에서, C/C++는 소프트웨어 코딩을 위하여 사용된다. 이러한 포팅은 클라이언트 측 포팅보다 상대적으로 쉽다. Windows NT에서 사용되는 각각의 API를 위하여, 이것들을 매칭하는 API(예: WinSock에서 BSD 소켓, Winthread에서 Pthread)가 Linux상에 있다. 그리고, 의미상의 작은 차이점들이 있다. 우리가 기억해야 할 것은 이러한 차이점들에 주의를 기울이는 것이다.

3. Java 코드화된 클라이언트/서버 애플리케이션. 이 포팅 방법은 아주 쉽다. Java 플랫폼이 독립적이기 때문에 변경 사항을 만들 필요가 거의 없을 것이다. 이 방법에 필요한 것은 귀사의 Linux상에 요구되는 모든 Java 클래스 라이브러리를 얻는 것이다.
4. n-Tier 아키텍처상의 미들 레이어. 이것은 ASP에서 JSP로, COM에서 EJB로와 같이 변경 사항을 요구할 수도 있다.
5. C/C++ 혹은 Java로 쓰인 소프트웨어 툴. 이러한 툴은 보통 싱글 워크스테이션 혹은 서버에서 사용되며 시스템상의 다른 구성 요소들과 상호작용을 거의 하지 않는다. 이러한 툴의 마이그레이션은 순수 Windows NT에서 Linux C/C++로의 포팅이다.

주의해야 하는 사항

이 가이드에서는, 일반 C/C++ 프로그램의 NT에서 Linux 포팅에 관한 토론 사항만을 제공한다. 그 이유는 이러한 경우가 가장 일반적이기 때문이다. C/C++에서는 특정 Window API와 Linux API의 차이점이 코드의 변화를 야기하는 이유이다. 그러나, 일반 운영체제에 의한 API는 아주 복잡하다. 여기서 우리는 이러한 차이점들에 관한 토론을 접기로 한다. 자세한 사항은 관련 Windows NT 및 Linux 문서를 참고 바란다. 아래의 리스트는 숙련된 포팅 프로그래머의 견지에서 바라본 고려 대상이다.

1. CPU 의존 문제점

- **머신 단어 길이 변화** 각기 다른 하드웨어 아키텍처는 머신 단어 길이를 가지고 있지 않다. Int 혹은 long은 항상 4바이트라고 단정하지 말라. 변수의 길이 혹은 오프셋과 연관 시, sizeof(..)를 사용하라.
- **Endian 타입의 차이점** 예를 들어, x86과 같은 몇몇 아키텍처는 small-endian이고, SPARC와 같은 몇몇 아키텍처는 big-endian이다. 두 아키텍처 사이에서 복합 데이터 구조가 서로 교환될 때, 사전 정의된 endian은 전송 매개로서 사용되어야 한다. 예를 들어, 애플리케이션 프로토콜을 원거리 노드로 전송하고 프레임 헤더가 많은 "int"

데이터를 가지고 있을 때, 프레임 전송 전에 이러한 데이터를 Network 바이트 순서로 변환해야 하며, 데이터 전송 후에는 데이터를 Network endian에서 로컬 endian으로 바꾸어야 한다.

2. 운영체제 차이점

- **파일 시스템 액세스 인터페이스 간의 차이점** Windows NT상에서, 각각의 물리적 파일은 디스크 C, D, E 등(논리 구획)과 같이 할당되며, 각각의 디스크 파일은 각 디스크 첫 번째 문자로 시작하는 완전 경로로 구분된다(예: C:\download\ guide.pdf). Linux 상에서는 각각의 파일 시스템 파티션이 독특한 루트 파일 시스템에 직접적 혹은 간접적으로 마운트된다. 그러므로 모든 디스크 파일들은 항상 유니폼 루트에서 시작하는 완전 경로를 통하여 구분된다. NT에서 Linux 포팅 시, 파일 구별에 관련된 코드들은 전부 변경되어야 한다.
- **내부-프로세스 커뮤니케이션** Windows NT 프로세스 간의 상호작용은 메시지-구동형이다. 하나의 Windows NT 과정은 외부 메시지들을 수용하여 다른 프로세스에 영향을 받기 위하여 메시지 대기열에 반복적으로 폴(poll) 하여야 한다. Linux에서도 또한 메시지 대기열 커뮤니케이션을 위한 메커니즘이 있다. 그러나, 그것은 선행조건이 아니다. 그리고, Linux는 순수 비동기 커뮤니케이션 방법인 Signal을 가지고 있다.

3. GUI 차이점

- Windows GUI 및 Linux GUI의 구축은 완전히 다르다. Windows NT GUI 애플리케이션은 Linux에서 GTK 혹은 QT 등의 GUI 라이브러리를 사용하여 다시 쓰여져야 한다.

4. 다른 차이점들

- **API상의 차이점** 이러한 타입의 종류는 보통 엄청나다. 그리고, 똑같은 업무를 위한 도입 방법, 기능의 이름 혹은 파라미터 포맷의 차이에 의해 표현된다. 예를 들어, Windows NT에서 파일을 열기 위해서는 OpenFile()이 사용된다. 반면, Linux에서는
-

open() 혹은 fopen()이 사용된다.

- **개발 또는 컨트롤 툴** Window NT에서 프로젝트 셋업 시 일반적으로 Makefiles이 보통 IDE에 의해 자동으로 생성된다. 그러나, 이러한 Makefile은 Linux에서 작동하지 않을 수도 있다. GNU가 Linux상에서 올바르게 인식하기 위하여 Makefile을 변경해야만 한다.
- **개발 혹은 컨트롤 툴** Windows NT에서 Visual SourceSafe는 버전 컨트롤을 위해 사용된다. 그러나, Linux상에서는 버전 컨트롤로 CVS가 주로 사용된다. 그리고 이것은 GUI 기반이 아니다. 그러므로, 위의 사실은 Windows NT 프로그래머들이 이러한 툴에 적응하는 데 약간의 시간을 필요로 할 수 있다.
- **Line end 바이트 차이점** 이것은 그다지 큰 문제점이 아닐 수도 있다. 그러나 이것은 모든 소스 파일들을 NT에서 Linux로 옮길 때 문제를 일으킬 수도 있다. Dos/Windows상에서, 텍스트 파일상의 Line end는 '\n\rd'의 2바이트로 마크된다. 그러나 Linux/Unix에서는 "\n"의 오직 1바이트 밖에 없다. Windows 소스 파일들을 Linux로 옮길 때, 이들 파일들은 몇몇 에디터 툴이 정확하게 인식하지 못할 경우도 있다. 각각의 NT 소스 파일들을 Linux 포맷으로 변경하기 위하여 "dos2unix" 툴을 사용할 수 있다.
- **컴파일러 차이점** Linux상의 GCC는 Window NT상의 Virtual C++와 상당히 많은 차이점을 가지고 있다. 그 예로, GCC는 케이스 민감하지만, VC++는 그렇지 않다. 성공적으로 링크된 C 프로그램은 Linux와 연결되지 않을 수도 있고, "미해결 심벌"과 같은 메시지를 받을 수 있기에 혼란스러울 수 있다. 이것은 원래 동일한 변수/기능이 케이스 차이로 인하여 GCC에 의해 다르게 보여지기 때문이다. 컴파일러의 차이점을 보여주는 또 다른 예는 타입 변경 식이다. Virtual C++에서 타입 변경 식은 "unsigned long (i)"와 같이 될 수 있다. 그러나 GCC에서는 "(unsigned long) i"만이 사용 가능하다.

포팅 과정

1. 소스 코드를 Linux로 옮겨라. GNU상에서 수행될 수 있도록 애플리케이션 Makefiles를 변경하라.
2. 코드와 헤더 파일을 포함하는 것들을 정리하라. 하드웨어 아키텍처에 심하게 의존하는 코드 포인트와 비형식 코딩 방법들을 제거하라.
3. 모든 경고 표시들을 켜놓은 상태에서 코드를 반복적으로 컴파일하라. 전체 소스 트리가 올바르게 컴파일되고 경고 혹은 에러 메시지가 뜨지 않을 때까지 경고와 에러를 교정하라.
4. 실행 가능한 애플리케이션을 실행하라. 만약 프로그램이 시작하지 않거나 곧 충돌하면, 보통 NULL 포인터 레퍼런스, 세그먼트 장애 등과 같은 명백한 문제점들이 있는 것이다. 문제점들을 빠르게 수정하고 3단계로 이동하라.
5. 만약 프로그램이 문제없이 실행되는 것 같다면, 테스터에게 애플리케이션 검사를 위하여 애플리케이션을 건네준다.

*from
NT
to
Linux*

포팅 툴

1. **Mainwin(<http://mainsoft.com/products/mainwin.html>)** Mainwin은 Windows 플랫폼상에서 Unix/Linux 코드를 컴파일하는 개발 툴이다. 이러한 툴은 마이그레이션을 훨씬 수월하게 하며 마이그레이션에 걸리는 시간을 현저하게 감소시킨다. Mainwin은 Visual Studio Environment에 심리스 통합(seamless integration)되어 있다.
 2. **EditPad(<http://www.editpadpro.com>)** Editpad는 Linux상에서 Windows 스타일의 편집 환경을 제공하는 강력한 편집 툴이다. 이것은 Vi와 같은 기존 Unix 에디터에 익숙하지 않은 장기간의 Windows 프로그래머들에게 특히 적당한 프로그램이다. 이러한 프로그램 역시 시간을 많이 단축시킨다.
 3. **Migratec(http://www.migratec.com/MigraTEC/migration_suite.htm)** Migratec은 마이그레이션에 아주 유용한 툴이다. 이 툴은 Linux상에서 바뀌어야만 하는 Window NT 소스 코드들을 표시해줄 수 있다.
-

2.4

검사

18
+
19

마이그레이션된 애플리케이션이 비즈니스에 공식적으로 구축되기 전에 애플리케이션의 검증이 이루어져야 한다. 검증 과정은 애플리케이션이 본래 규격에 정의된 요구 사항들을 만족하는지의 여부를 확인하는 과정이다. 검증 과정은 오랜 시간에 걸친 공식적인 검사에 의해 이루어진다. 마이그레이션된 애플리케이션을 위한 검사는 검사가 새로운 플랫폼에서 수행되어야 하는 경우를 제외하고는 새롭게 개발된 애플리케이션의 검사와 크게 다를 바 없다. 검사는 보통 마이그레이션된 애플리케이션의 기능성뿐만 아니라 성능 평가 또한 포함한다. 검사 작업을 용이하게 하기 위하여 검사 틀이 주로 사용된다. 몇몇 검사 틀은 대상 플랫폼상에서 실행을 요구한다. 이러한 종류의 검사 틀은 보통 Java, Perl 등과 같은 고이식성 언어(portable language)로 쓰여 있다. 그렇지 않으면 틀을 대상 플랫폼에 우선 포팅해야 할지도 모른다.

검사는 아래의 종류로 나눌 수 있다.

- 설치 검사
- 기능성 검사
- 호환성 검사
- 성능 검사

검사 도중 애플리케이션이 논리적으로 실행이 안 되거나, 심지어 뜻밖에 종료됨을 경험할 수 있다. 이러한 경우, 문제점이 어디에 있는지 찾기 위하여 코드를 디버그 해야만 한다. GDB는 Linux에서 가장 유용한 C 프로그램 디버깅 틀이며 사용 방법 또한 간편하다. GDB를 사용하기 위하여, 모든 C 소스 파일들을 단순히 “-g” 옵션과 함께 컴파일 하면 된다. Linux Shell 상에서 “info gdb”라고 입력하면 GDB 사용에 관한 모든 정보를 얻을 수 있다.

2.5

성능 튜닝

1

• 2

3

4

애플리케이션의 성능을 튜닝하는 것은 고도로 숙련된 작업이라 할 수 있다. 아래의 내용은 C/C++ 프로그램의 성능을 향상할 수 있는 여러 방법들을 기술하고 있다.

1. 시스템 호출 감소. 시스템 호출은 유저 스페이스에서 커널 스페이스에 이르는 애플리케이션을 트랩하기 때문에 시간을 소비하는 작업이다. 만약 작업이 유저 레벨로 구축될 수 있다면, 시스템 호출을 피하면서 유저 레벨로 구축하도록 한다.
2. 기능 호출 보호. 기능 보디가 아주 작으면, 기능 호출과 연관된 오버헤드는 간과되지 않을 수도 있다. 이런 이유로, 가능하다면 매크로 혹은 인라인 기능들을 사용하라.
3. 코드 포인트 발견을 위해 프로그램들을 프로파일 하라. sprof와 gprof가 Linux상에서 사용 가능하다. sprof는 매번 단일 다이내믹 라이브러리의 기능들만을 프로파일 할 수 있다. gprof는 프로그램 전체를 프로파일 할 수 있지만, 소스 파일들을 “-pg” 옵션과 함께 컴파일 해야만 한다. 위 툴의 사용법을 알고 싶다면 Linux 툴에 관한 매뉴얼 페이지를 참조하라.
4. 메모리 프로파일. 단일 애플리케이션의 메모리 사용 프로파일을 위하여 Linux상의 Memprof 툴을 사용할 수 있다. Memprof는 애플리케이션을 실행함에 따라 메모리 할당 활동의 변화를 보여주는 히스토그램을 출력한다. 이 툴은 다이내믹 메모리의 불합리한 할당/반환의 가능한 경우들을 찾는 데 유용하다.
5. 단축 평가(short-cut evaluation)에 주의하라. || 및 && 조건식은 코드 실행에 있어 순차적이다. 컴파일러가 식을 왼쪽에서 오른쪽으로 평가한다는 가정하에, 오른쪽 || 의 식이 1이 될 가능성이 아주 크다면 이것을 왼쪽에도 넣어 코드의 능률성을 향상시킬 수 있다는 것이 명백하다. 이와 비슷하게, &&의 오른쪽 식이 0이 될 가능성이 아주 높다면 이것을 왼쪽에도 입력하여 코드의 능률성을 향상시킬 수 있다. 식을 넣는 방법을 알게 되어, 그 결과 더욱 좋은 코드들을 만들어낼 수 있다는 것을 명심하라.

*from
NT
to
Linux*

2.6

구축

20

+
21

애플리케이션 구축 과정은 아래의 과정을 포함한다:

1. 새로운 런-타임 환경 구축

- 하드웨어 장비를 셋업한다. 이것은 마이그레이션 구축이기 때문에 기존의 물리 장비는 바뀌지 않아도 된다.
- 대체될 NT 서버에 Linux를 설치한다.
- 기존 플랫폼상의 기존 서비스를 대체하는 새로운 표준 서비스를 새로운 플랫폼상에 셋업한다. 이것은 파일 & 프린팅 서비스, 웹 서비스, 파이어월, 데이터베이스 서버, 애플리케이션 서버 등과 같은 것을 포함할 수 있다.
- 마이그레이션된 애플리케이션 런-타임 소프트웨어를 새로운 플랫폼상에 설치한다.

2. 기존 시스템에서 새로운 시스템으로 데이터 마이그레이션

기존 시스템의 데이터를 마이그레이션하는 것은 비즈니스 일관성 관점에서 필요하다. 보통 이 과정은 기존의 데이터베이스(예: MS SQL)에서 애플리케이션 데이터의 백업/익스포트 작업과 새로운 데이터베이스 시스템(예: DB2)으로의 복원/임포트 작업을 포함한다.

3. 애플리케이션 실행

3

1

2

• 3

4

미들웨어 마이그레이션 전략은 Windows에만 편향된 미들웨어 기반 애플리케이션을 고도로 개방되고 유연성을 갖춘 IBM 미들웨어로 옮기는 데 추천되는 접근법이다. IBM 미들웨어 소프트웨어 플랫폼은 Linux, Windows, AIX, OS/400 및 z/OS와 같은 다양한 운영체제에서 실행이 가능하다. Linux상의 IBM 미들웨어는 Intel 기반 데스크톱/서버에서부터 POWER 기반 워크스테이션뿐만 아니라, 강력한 메인프레임에 이르는 다양한 하드웨어 플랫폼 상에서도 구동이 가능하다. 마이그레이션 과정은 지금보다 훨씬 넓은 애플리케이션의 확장성 옵션을 가능하게 할 것이며 미래에 요구되는 마이그레이션을 최소화할 것이다.

*from
NT
to
Linux*

Middleware Migration

미들웨어 마이그레이션

이번 장에서는 MS Exchange 서버에서 Domino 마이그레이션, SQL 서버에서 DB2마이그레이션에 대하여 논의해 보겠다. 그 밖의 미들웨어 마이그레이션 가이드는 추후 요청 시 추가 예정이다.

3.1

MS Exchange 서버에서

Domino 마이그레이션

1

2

• 3

4

사전 평가

전제 조건:

1. 트레이닝 혹은 그 밖의 방법으로 Domino 관련 기술을 채득하라.

from
NT
to
Linux

포팅 분석

1. Domino 마이그레이션
 - a) POP3/IMAP/SMTP 서비스 마이그레이션
 - b) 유저 메일박스
 - c) 공용 폴더
 - d) 기타...
2. 클라이언트 마이그레이션
 - a) 유저 수
 - b) 유저 메일박스 크기
 - c) 메일 클라이언트를 Outlook에서 Lotus Notes로 마이그레이션
 - d) 기타...

참:

Redbook: Migrate Exchange 5.5 for NT to Domino on Linux

리소스:

<http://www.redbooks.ibm.com/>

<http://www-1.ibm.com/servers/solutions/collaboration/move2lotus/>

Migrate Exchange 5.5 to Domino on Linux, REDP-3777

Exchange Your Exchange: Migrating from Microsoft Exchange 5.5 to Lotus

Domino 6, REDP-3694

24
+
25

포팅

전제 조건:

1. 제품 권고
2. Domino 준비 및 설치
3. 자동화 서버 구조 마이그레이션(유저 및 그 밖의 서버 데이터)과 유저 메일박스 마이그레이션(예: Lotus Notes and Domino Admin, Outlook)에 사용될 방법 및 툴 결정
4. Lotus Notes, Domino Admin, Outlook 준비 및 설치
5. Lotus Notes와 Domino Admin 사용 환경에 익숙함

포팅 과정:

1. 유저 및 그 밖의 서버 데이터 마이그레이션
 - a) 메일박스
 - b) 폴더 구조가 있는 Inbox 보기
 - c) 메일 메시지 변환
 - d) 캘린더 엔트리
 - e) 연락처
 - f) 업무/할일 엔트리
2. 공용 폴더 마이그레이션
3. 유저 메일 클라이언트 마이그레이션

툴:

1. 유저 및 그 밖의 서버 데이터의 마이그레이션을 위하여 아래의 사항이 요구된다.
 - a) DUS(Domino Upgrade Services)를 갖춘 Domino Administrator 클라이언트
 - b) Outlook 클라이언트
 2. 공용 폴더 마이그레이션을 위하여 아래의 사항이 요구된다.
 - a) Exchange Administrators 프로그램
 - b) Outlook 클라이언트
 - c) Notes 클라이언트
-

1

2

• 3

4

d) Excel 혹은 Lotus 123

from
NT
to
Linux

Domino Upgrade Services(DUS)

Domino Administrator 클라이언트 상의 DUS 마이그레이션 툴은 기존 Microsoft Exchange의 메시지 및 디렉토리 시스템의 정보들을 Domino 6로 쉽게 마이그레이션해 주는 툴이다. DUS를 가지고, Exchange Global Address List와 같은 외부 디렉토리로부터 유저들을 임포트할 수 있고 Notes 유저로 등록한 다음 Exchange 메일 시스템의 메일들을 Domino 메일 데이터베이스로 변경할 수 있다. 이 마이그레이션 툴은 외부 디렉토리의 모든 유저들의 마이그레이션을 가능하게 해주며 혹은 특정 유저만을 골라 마이그레이션할 수 있게 해준다. 이 툴은 Domino Administrator에 포함되어 있다.

리소스:

Migrating from Microsoft Exchange 5.5 to Lotus Notes and Domino 6, SG24-6955
The Three Steps to Super.Human.Software: Compare, Coexist, Migrate From Microsoft Exchange to Lotus Domino Part One: Comparison SG24-5614
The Three Steps to Super.Human.Software: Compare, Coexist, Migrate From Microsoft Exchange to Lotus Domino Part Two: Coexistence and Migration SG24-5615

검정 및 검사

전제 조건:

1. 계획의 목적 열거, 검사 계획의 레벨 파악
2. 검사의 범위 파악
3. 검사 툴의 획득 및 설치

검정 및 검사 과정:

1. 계획
-

26
+
27

- a) 범위
 - b) 검사 아이템의 정의
 - c) 검사받을 기능들
 - d) 아이템 합격 여부 기준
 - e) 정지 기준 및 재개 요구 사항
 - f) 검사 인도
 - g) 환경 요구 사항
2. 서버 및 클라이언트 검사
- a) 메일 서버 업무 체크
 - b) 유저 메일박스 구성 체크
 - c) 유저 메일박스 기능 체크
3. 문제 해결
- a) 관련 메시지 체크
 - b) 에러 코드 파악
 - c) 문서화 체크
 - d) 인터넷 리소스를 통한 검색
 - e) 문제 폭을 좁히기 위한 툴의 사용

툴:

IBM Rational Suite TestStudio

Domino NSD tool - Notes System Diagnostic(NSD) 등...

리소스:

<http://www.ibm.com/software/rational>

1

2

• 3

4

성능 튜닝

전제 조건:

1. Domino 아키텍처와 고급 Domino Administration 기술들을 익힌다.
2. 검사 데이터를 Domino로 로드한다.

from
NT
to
Linux

성능 튜닝 과정:

1. OS 레벨 튜닝
 - a) 튜닝 시스템 캐시
 - b) 튜닝 시스템 네트워킹
 - c) IDE vs. SCSI
 - d) I/O 분산
 - e) glibc 업그레이드
 - f) 기타...
2. Domino 레벨 튜닝
 - a) 클러스터링
 - b) 트랜잭션 로깅
 - c) 다양한 메일박스
 - d) 기타...

툴:

1. Linux 네트워크 툴
2. Domino NSD 툴 - Notes System Diagnostic (NSD)
3. Linux상의 Quest Spotlight

레퍼런스:

IBM Redbooks

Migrate Exchange 5.5 nge 5.5 to Domino on Linux, REDP-3777

28
+
29

Exchange Your Exchange: Migrating from Microsoft Exchange 5.5 to Lotus Domino 6, REDP-3694

Migrating from Microsoft Exchange 5.5 to Lotus Notes and Domino 6, SG24-6955

The Three Steps to Super.Human.Software: Compare, Coexist, Migrate From Microsoft Exchange to Lotus Domino Part One: Comparison SG24-5614

The Three Steps to Super.Human.Software: Compare, Coexist, Migrate From Microsoft Exchange to Lotus Domino Part Two: Coexistence and Migration SG24-5615

Linux Handbook: A Guide to IBM Linux Solutions and Resources, SG24-7000

Linux, Why It Should Replace Your Windows NT Domain, REDP-3779

Lotus Domino 6 for Linux, SG24-6835

Open your Windows With Samba, REDP-3780

온라인 리소스

Redbooks

<http://www.redbooks.ibm.com>

DeveloperWorks

<http://www.ibm.com/developerworks/cn>

Lotus 웹사이트

<http://www.lotus.com>

Domino 크기 조절에 관한 자세한 정보는 Domino 서버 용량 벤치마크를 측정하기 위해 디자인된 Server.load 툴을 참조:

<http://www-1.ibm.com/support/docview.wss?uid=swg21089092>

Tivoli 웹사이트, Tivoli Storage Manager

<http://www.ibm.com/tivoli>

NetWorker Module for Lotus Notes

<http://www.legato.com>

Symantec 웹사이트

<http://www.symantec.com>

3.2

SQL 서버에서 DB2 마이그레이션

1

2

• 3

4

왜 DB2로 포팅하는가?

DB2는 여러 기술에서 데이터베이스의 선두를 달리고 있으며, 트루 멀티-플랫폼 지원 및 확장성을 제공한다. 이 데이터베이스는 단일 서버에서 여러 종류의 워크로드를 처리할 수 있다. DB2는 훌륭한 성능을 유지하면서 대용량 온라인 트랜잭션 프로세싱(OLTP)에서부터 복잡한 멀티-유저 쿼리 등과 같은 다양한 워크로드를 처리한다.

확장성 및 성능 이외에도, DB2는 아래와 같은 장점을 제공한다.

- 네이티브 환경의 통합 지원
- 통합 시스템 관리 툴과 멀티-플랫폼 툴
- 자가-관리 및 리소스 튜닝 성능
- 데이터 복제 서비스
- 통합 웹 액세스
- 웹 서비스 애플리케이션
- 개발 환경의 통합 지원
- 데이터 웨어하우스 기능
- 고가용성 지원
- 개발자 및 지원을 위한 IBM 프로그램

사전 평가

전제 조건:

훈련 혹은 그 외의 방법을 통하여 DB2 기술 취득

포팅 분석:

데이터베이스 마이그레이션

1. 데이터 종류
2. SQL 기능
3. 인덱스
4. 축적 절차

*from
NT
to
Linux*

30
+
31

5. 외부 결합
6. 커서
7. 글로벌 임시 테이블
8. SQL 서버 계산형 컬럼
9. 기타...

애플리케이션 마이그레이션

1. ADO 및 OLE DB
2. ODBC 및 CLI
3. 임베디드 SQL
4. JDBC
5. 기타...

툴:

DB2 Porting Assessment Guide

포팅

전제 조건:

1. 제품 권고
2. DB2 UDB의 획득 및 설치
3. 자동화 데이터 및 데이터베이스 변환에 사용될 방법 및 툴 결정(예: IBM Migration Toolkit)
4. Migration Toolkit의 획득 및 설치
5. Migration Toolkit의 사용 환경에 친숙함

포팅 프로세스:

데이터베이스 구조의 포팅

데이터베이스 객체의 포팅

1

2

• 3

4

-
1. 축적 절차
 2. SQL 기능
 3. 트리거
 4. 기타...

from
NT
to
Linux

애플리케이션 변경

1. CLI 및 ODBC
2. ADO 및 OLEDB
3. JDBC
4. 임베디드 SQL
5. 데이터 마이그레이션

툴:

IBM DB2 Migration Toolkit(추천됨)

IBM DB2 Migration Toolkit는 지원되는 모든 DB2 UDB 워크스테이션 플랫폼상에서 SQL 서버 데이터베이스를 DB2 UDB 데이터베이스 마이그레이션을 도와주는 툴이다. 이 툴은 테이블, 인덱스, 일차 키, 보기 및 축적 절차를 만드는DDL 스크립트를 생성하는데 사용될 수 있다. 이 툴은 DB2 Migration Home에서 다운로드 받을 수 있다.

검정 및 검사

전제 조건:

1. 계획의 목적 열거, 검사 계획의 레벨 파악
2. 검사의 범위 파악
3. 검사 툴의 획득 및 설치

검정 및 검사 과정:

1. 계획
 - a) 범위
 - b) 검사 아이템의 정의
-

32
+
33

- c) 검사받을 기능들
 - d) 아이템 합격 여부 기준
 - e) 정지 기준 및 재개 요구 사항
 - f) 검사 인도
 - g) 환경 요구 사항
2. 데이터 체크 기술
- a) 코드 및 애플리케이션 검사
 - b) 보기 정상 감시 체크
 - c) SQL PL 객체 체크
 - d) 애플리케이션 코드 체크
3. 문제 해결
- a) 관련 메시지 체크
 - b) 에러 코드 설명
 - c) 문서화 체크
 - d) 인터넷 리소스를 통한 검색
 - e) 문제 폭을 좁히기 위한 툴의 사용

툴:

IBM Rational Suite TestStudio
DB2 네이티브 툴: 스냅샷 모니터링
이벤트 모니터링
기타...

성능 튜닝

전제 조건:

1. DB2 아키텍처와 고급 DB2 Administration 기술들을 익힌다.
 2. 검사 데이터를 데이터베이스로 로드한다.
-

1

2

• 3

4

성능 튜닝 프로세스:

OS 레벨 튜닝

1. Kernel 파라미터
2. 튜닝 시스템 캐시
3. Raw 디바이스 재할당
4. 기타...

*from
NT
to
Linux*

DBM 레벨 튜닝

1. 백업 버퍼 사이즈
2. 버퍼 사이즈 복원
3. RQRIOBLK
4. 데이터베이스 객체의 물리적 배치
5. 기타...

DB 레벨 튜닝

1. 버퍼 풀
2. Db Heap
3. Sort Heap
4. IO 클리너 숫자
5. IO 서버 숫자
6. 기타...

환경 변수(db2set 커맨드):

1. DB2_RR_TO_RS
2. DB2_MMAP_WRITE
3. DB2_MMAP_READ
4. DB2_FORCE_FCM_BP
5. DB2_PARALLEL_IO
6. 기타...

참고:

Quest Central for DB2

34
+
35

Db2 native tools:
db2batch
dynexpln
Configuration Advisor
Index advisor
기타...

자세한 정보는 아래의 사이트를 참조

- DB2 제품군
www.ibm.com/software/data/db2
 - DB2 제품 및 서비스 기술 라이브러리
www.ibm.com/software/data/db2/library
 - DB2 유지/보수
www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report
<ftp://ftp.software.ibm.com/ps/products/db2/fixes/<language>/<platform>/<FixPak#>>
 - 데이터 관리 기술 페이지
www.ibm.com/software/data/db2/skills/
 - DB2 웹 페이지
www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/db2web.d2w/report
 - DB2 마이그레이션 웹사이트
www.ibm.com/software/data/db2/migration
 - DB2 개발자 도메인 라이브러리
www.ibm.com/software/data/developer
 - DB2 Today
www.ibm.com/software/data/db2today/
 - 개발자 프로그램을 위한 IBM PartnerWorld
www.developer.ibm.com/
-

4

1

2

3

• 4

*from
NT
to
Linux*

NT Services Migration

NT 서비스
마이그레이션

4.1

마이그레이션 고려 사항

	1
	2
	3
•	4

마이그레이션이 가능한 서비스들은

- 유저 및 도메인 인증 서비스
- 파일 및 프린팅 공유 서비스
- 그 밖의 기본 네트워크 서비스

*from
NT
to
Linux*

NT 서비스를 Linux 마이그레이션하기 위하여 해야 할 많은 일들이 있다. 첫 번째 해야 할 일은 어떠한 서비스를 Linux 마이그레이션할 것인지 정의하는 것이다. 또한 Linux 마이그레이션을 수행하기 전에 NT 시스템에서 충분한 정보를 수집해야만 한다. 준비 과정이 끝난 후, NT에서 Linux 마이그레이션에 관한 올바른 방법을 발견할 수 있다.

마이그레이션 과정

마이그레이션 준비

- NT 시스템에서 필요한 정보 수집
- 유저 인증 정보 수집
- 구성 정보 수집
- 서비스 데이터 수집

마이그레이션

- Linux에 이에 상응하는 서비스 설치
- 유저 인증 정보를 시스템으로 импорт
- 서비스 데이터를 Linux로 이동

아래의 내용은 서비스 마이그레이션에서 알아야 하는 고려 사항이다.

4.2

유저 인증

38
+
39

Window NT 환경에서, 유저들은 항상 특정 도메인으로 로그인해야만 한다. 이러한 경우, NT 서비스를 Linux 마이그레이션하는 것은 NT상의 유저 인증을 Linux 마이그레이션하는 것을 포함할 것이다. Samba는 Linux 상에서 이러한 유저 인증 기능을 제공하는 서비스 구성 요소이다.

Samba는 Windows 및 그 밖의 클라이언트에 Linux상의 PDC(Primary Domain Controller) 기능을 제공한다. 유저 인증 및 Samba 도메인을 구성하기 위하여 구성 파일인 `/etc/smb.conf`를 수동으로 변경해야만 한다. 또한 웹 운영 GUI가 포함된 Samba의 구성을 원한다면 SWAT 패키지를 시스템에 설치해야 한다.

4.3

파일 및 프린트 공유 서비스

	1
	2
	3
•	4

기업 인트라넷의 주요 기능은 파일 및 프린트 공유 서비스이다. 대부분의 클라이언트는 Windows 제품이기 때문에, 기업들은 항상 파일 및 프린트 공유 서비스를 제공하기 위하여 NT 서버를 사용한다. Samba는 Linux 환경에 이러한 서비스 제공을 위하여 사용된다.

*from
NT
to
Linux*

대부분의 Linux 배포 업체들은 이미 Samba 패키지를 번들화했기 때문에 Linux 시스템을 설치 시 Samba 패키지를 선택하기만 하면 된다. Linux 설치 시 Samba를 설치하지 않았다면, Linux RPM 패키지에서 추후 설치가 가능하다. 파일 및 프린트 공유 서비스와 Samba 도메인을 구성하기 위하여, 구성 파일인 `/etc/smb.conf`를 수동으로 변경해야 한다. 웹 운영 GUI가 포함된 Samba의 구성을 원한다면 SWAT 패키지를 시스템에 설치해야만 한다.

4.4

그 밖의 기본 네트워크 서비스

40
+
41

마이그레이션이 가능한 서비스들은 아래와 같다

- DHCP
- DNS
- FTP
- HTTP

거의 대부분의 서비스들이 Linux 배포판에 번들로 제공되기 때문에 Linux 설치 시 이러한 서비스들을 단순히 선택할 수 있다.

NT 서비스와 Linux 서비스의 비교

NT 서비스	Linux 서비스	비고
DHCP	dhcp	
DNS	bind	
FTP (IIS)	vsftp./ wu-ftp / proftp	
HTTP (IIS)	Apache	IIS상의 ASP 페이지는 Apache상의 PHP 페이지로 전환 요망
파일 및 프린트 공유 서비스	Samba	웹 운용관리 GUI의 사용은 SWAT 패키지 설치 필요
유저 인증 서비스	Samba	웹 운용관리 GUI의 사용은 SWAT 패키지 설치 필요



© International Business Machines Corporation 2004

서울시 강남구 도곡동 467-12
군인공제회관빌딩

한국아이비엠주식회사
고객만족센터

TEL: (02)3781-7114
www.ibm.com/kr

Printed in Korea
All Rights Reserved

IBM, IBM 로고, e-business 로고, zSeries, pSeries, iSeries, xSeries, DB2, Lotus, Tivoli는 International Business Machines Corporation의 상표 또는 등록상표입니다.

Linux는 Linus Torvalds의 등록상표입니다.

기타 회사, 제품과 서비스명은 기타 회사의 등록상표 또는 서비스 마크입니다.

본 자료에 사용된 제품의 사진은 출하시의 제품과 다소 차이가 있을 수 있습니다. 또한 여기에 실린 내용 및 제원은 사전 예고 없이 변경될 수 있습니다.

—
www.ibm.com/kr/linux