[기술자문 컨설팅]

과학기술부 Solaris의 Linux 마이그레이션

한국소프트웨어진흥원

공개SW기술지원센터

<Revision 정보>

일자	VERSION	변경내역	작성자
2007.11.07	0.1	초기 작성	김용규

목 차

1. 문서 개요

본 문서는 고비용의 Unix계열중 Solaris의 공개SW 기반 OS인 Linux로의 마이그레이션 방안에 대한 권고 가이드로 작성되었으며, 과학기술부에 기술자문 컨설팅 참고자료 활용을 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- OS 마이그레이션 수행 시 고려사항
- OS 마이그레이션 수행 절차
- 상용 Solaris에서 공개SW Linux로의 구축방안

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- 공개SW로의 OS 마이그레이션 수행시 컨설팅 참고자료로 활용할 수 있다.
- 상용 Solaris에서 공개SW Linux로의 OS 마이그레이션 구축방안과 유지보수에 대한 참고자료로 활용 할 수 있다.

2. 대상기관 현황 분석

가. 대상 기관

기관명*	과학기술부	웹사이트	http://www.most.go.kr/
주소*	경기도 과천시 관문로 88(중앙동) 정부과천청사 과학기술부		
연락처*	02-504-0333	E-MAIL*	
분석 자료	공개SW기반 마이그레이션(Solaris->Linux) 방안		

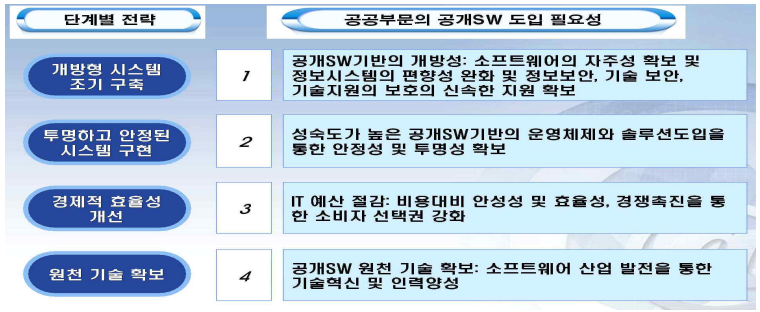
나. 분석 대상

과학기술부의 마이그레이션 고려시 참고자료로 활용하기 위해 작성하였고, 이를 통해 공개SW 및 국산SW 도입을 위한 방안을 제시하기 위한 마이그레이션 컨설팅 참고자료로 작성하였다.

3. 공개SW마이그레이션 준비단계

가. 공개SW 마이그레이션 단계별 전략

공개SW 마이그레이션을 수행함에 있어 사업의 목적이 달성될 수 있는 가치를 분석하고 이에 대한 적절한 IT전략이 필요하다고 하겠다. 다음은 단계별 전략을 도출한 내용이다.



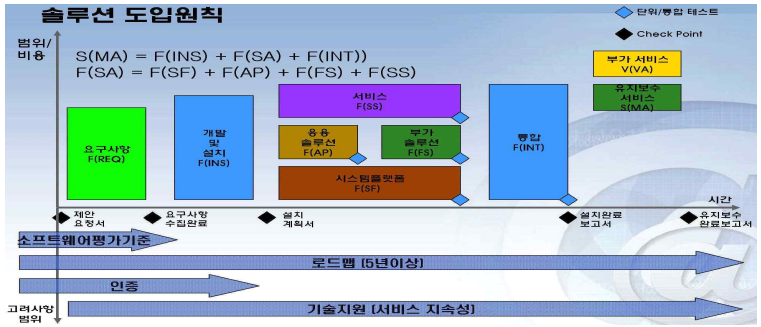
<그림 1 공개SW 마이그레이션 단계별 전략>

나. 공개SW 마이그레이션시 도입원칙

1) 로드맵
- 도입시점 기준 최소 5년이상의 해당솔루션에 대한 로드맵 보유업체

2) 인증(Certification)
- 도입예상 솔루션간 인증 보유(사전인증/사례인증)
- 하드웨어/운영체제/솔루션1/솔루션2...

3) 기술지원(Technical Support)
- 기술지원 및 유지보수 전국망 보유
- 지역내 기술지원 및 유지보수 망 보유



<그림 2 공개SW 마이그레이션시 도입원칙>

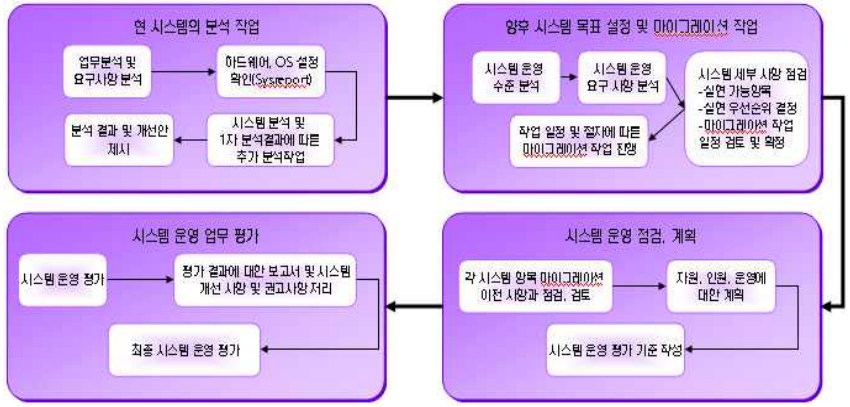
다. 공개SW 도입 단계별 고려사항

1) 사업계획서 작성 단계
가) 시스템 개발 시 개방 표준(Open Standard. ISO, W3C 등 국제표준기구 혹은 단체에서 수용한 기술 표준), 개방형 플랫폼(Open Platform. 개방 표준에 부합하는 소프트웨어 혹은 하드웨어), 상호호환성을 가지는 제품을 우선 고려할 것을 권고하도록 한다.
나) 정보화 예산 편성 지침 및 연도별 세출 예산 편성 지침 등의 공개소프트웨어 관련 내용을 숙지하여 사업계획서를 작성하도록 한다.

2) 제안요청서 작성 단계
가) 제안요청서 작성 시 공개소프트웨어 도입을 저해하는 비표준적인 특정 기술조건을 명시하지 않도록 한다.
나) 시스템 계층별 독립성을 확보하기 위해서 제안요청서 상의 '도입대상 장비내역 및 구성요건' 작성 시, 하드웨어와 운영체제를 별도의 항목으로 명시하고 별도의 비용으로 계상하도록 하며, 응용소프트웨어와 운영체제를 별도의 항목으로 명시하고, 별도의 비용으로 계상하도록 한다.
다) 포털 및 웹 기반 서비스 구축의 경우 주관기관은 제안요청서 작성 시 국민의 정보접근권리 보장을 위하여 '다양한 컴퓨팅 환경에서 접근이 가능하도록 국제표준을 준수하는 제품 도입 및 개발'에 대한 항목을 명시하도록 한다.
라) 공개소프트웨어 기반 환경의 웹 정보시스템 도입 및 구축을 계획할 때에는 반드시 정보 시스템의 상호 운용성 확보를 명기한다(정보시스템 구축 운영 가이드라인*의 체크리스트를 이용하여 사업계획서 작성 시 준수할 각종 표준에 대해 명시하도록 하고, 사용자 인터페이스 부분의 준수를 명시하도록 한다).
마) BPR/ISP 사업의 제안요청서 작성 시 공개소프트웨어 적용 가능성 분석을 포함시키도록 한다.
바) '공개소프트웨어를 제안하는 사업자는 사업 완료 이후 공개소프트웨어 유지보수 방안 제시' 에 대한 항목을 명기하도록 한다.
사) 개별 대상 업무 응용소프트웨어 납품 시 포함될 수 있는 오픈소스를 포함한 소스코드의 명세서에 대한 사용 형태 및 정도, 라이센스 의무사항 준수 여부와 미 준수사항을 제출하도록 의무화한다.

3) 도입 평가 단계
가) 주관기관은 소프트웨어 선택 시 동등한 성능일 경우 공개소프트웨어를 제안한 업체를 우선적으로 고려하도록 한다.
나) 제안서 평가 시 공개소프트웨어 도입 여부를 평가 항목에 반영한다.

라. 공개SW 마이그레이션 수행절차



<그림 3 공개SW 마이그레이션 수행절차>

1) 현 시스템 분석작업
2) 향후 시스템 목표설정 및 작업
3) 시스템 운영점검, 계획
4) 시스템 운영업무 평가

4. 공개SW 마이그레이션 수행

가. 마이그레이션을 위한 사전 환경분석

1) Solaris에서 Linux로 마이그레이션 개요

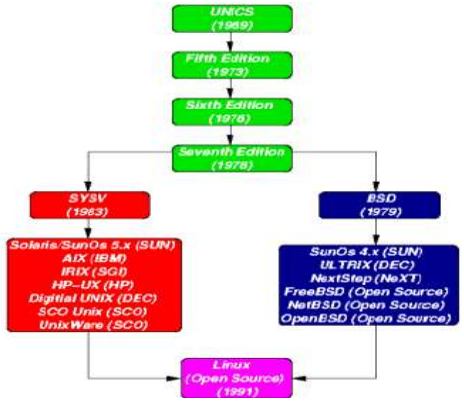
가) Linux 최신 커널 기반의 고객 지원 및 중요한 비즈니스 운영 환경에서 응답 시간이 단축되므로 어플리케이션 성능이 향상

나) 독점(proprietary) UNIX 시스템을 저비용의 상용 Intel 하드웨어로 대체함으로써 IT 운영의 총 비용이 절감

다) UNIX와 Linux는 기술적으로 서로 유사하므로 사용자가 이미 보유하고 있는 IT 기술을 적극 활용

라) Intel 플랫폼에서 시스템의 보안과 확장성이 향상

2) Solaris / Linux 비교



<그림 4 유닉스/리눅스 역사>

가) 유사점

(1) Kernel : 대부분의 BSD, SYSV system calls를 구현하고 있으며, POSIX.1을 따름

(2) Shell : 유니긋 시스템과 거의 유사

(3) System Utility : 유닉스 시스템과 유사

(4) Application : gcc, g++, javac, vi, emacs, make, gdb/dbx, perl, etc

나) 리눅스의 장점

(1) 대부분의 하드웨어를 지원

(2) 기술발전이 빠르다.

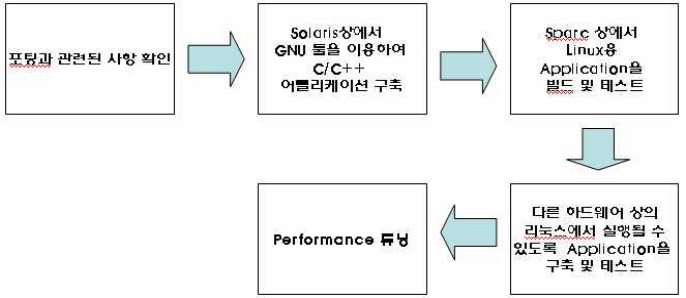
(3) 저렴한 비용으로 동급의 성능을 제공

다) 리눅스 Kernel 2.6 지원내역

항목	지원 여부	제 공
NPTL(Native Posix Thread Library)	Yes	High performance POSIX compliant multithreading
Kernel IPSec	Yes	IPSec layer available for use by kernel modules
Asynchronous I/O (AIO)	Yes	Improved application performance
O(1) Scheduler	Yes	Highly scalable SMP scheduler
kksymoops	Yes	Improved kernel bug reporting
Reverse Map Virtual Memory(rmap VM)	Yes	Performance improvement in memory constrained systems
HugeTLBFS	Yes	Performance improvement for large virtual memory applications (e.g. Databases)
Remap_file_pages	Yes	Kernel memory optimization for shared memory applications
2.6 Network stack features(IGMPv3, Ipv6, etc.)	Yes	Improved network performance & messaging
IPvs	Yes	Network load balancing
Access Control Lists (ACLs)	Yes	Improved file system security management
4GB-4GB memory split	No	Greatly increased x86 physical memory support and larger application address space
Scheduler support for No hyperthreaded CPUs	No	Improved hyperthreaded CPU performance. (2.6 implementation not yet comparable)
Block I/O (BIO) block layer	Yes	Major rewrite of the I/O subsystem (stabilization and driver support in progress)
Support for > 2TB file system	Yes	Support for very large volumes. Red Hat Enterprise Linux 3 supports up to 1 TB.
New I/O elevators	Yes	Fine tuning for I/O subsystem performance(stabilization in progress)
Interactive scheduler response tuning	Yes	Scheduler improvements for interactive tasks(stabilization in progress)

<표 1 리눅스 커널 2.6의 지원내역,2007현재>

나. 어플리케이션 마이그레이션 방법론



<그림 5 어플리케이션 마이그레이션 방법론>

1) 포팅과 관련된 사항 확인
   가) 시스템 관리
      (1) 일반적으로 리눅스 관리는 다른 유닉스 운영체제의 관리와 유사

   나) 소스 코드 관리
   다) 기타 ThirdParty Tool, 유틸리티, 라이브러리
      (1) Solaris 상에서 일반적이고 대중적인 써드파티 툴, 유틸리티와 라이브러리들은 리눅스에서도 사용
   라) 64비트 컴퓨팅
      (1) 레드랫과 수세 및 기타 많은 배포본들은 64비트 리눅스 버전 제공
   마) [1]Endian Format
      (1) SUN 스팍과 울트라 스팍 프로세서는 Big Edian 형식으로 정수를 저장
      (2) 어플리케이션 내의 Endian포맷에 대한 의존성을 해결

2) Solaris 상에서 GNU 툴을 이용하여 C/C++ 어플리케이션 구축
   가) 리소스
      (1) 솔라리스 GNU 툴
      (2) make 툴
         (가) GNU make
         (나) Sun make
      (3) 컴파일러 툴

1) Endian :단어를 형성하는 2진 바이트에서 저장하는 바이트의 순서를 나타내는 방법. 빅 엔디안(big-endian)과 리틀 엔디안(little-endian)이 있는데, 빅 엔디안은 최상위 비트(MSB)부터 부호화되어 저장되며, 리틀 엔디안은 최하위 비트(LSB)부터 부호화되어 저장된다. 예를 들면, 숫자 12는 2진수로 나타내면 1100인데 빅 엔디안은 1100으로, 리틀 엔디안은 0011로 각각 저장된다. -인터넷 네이버(naver.com)제공-

(가) GCC
(나) Sun C, Sun C++
나) 솔라리스 make 유틸리티 대신 GNU make 유틸리티를 이용하여 어플리케이션 구축
다) makefile을 gmake와 호환이 되도록 수정한 후에, C컴파일러 이름을 변경
라) 빌드할 때 나타나는 에러메시지를 명령행 옵션과 코드 문제로 구별하여 해결
마) 라이브러리가 속해있는 라이센스와 저작권을 확인

3) Sparc 상에서 Linux용 Application을 빌드 및 테스트
가) SUN 하드웨어에서 실행 가능한 리눅스를 설치
나) 제공된 GNU 툴을 이용하여 어플리케이션을 리빌드
다) 어플리케이션에 맞도록 수정
라) 전체적인 검증 테스트를 실행

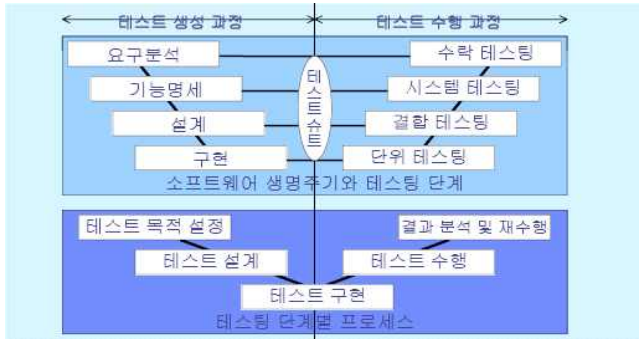
4) 기타 다른 하드웨어 상의 리눅스에서 실행될 수 있도록 Application을 구축 및 테스트
가) 포팅하고자 하는 하드웨어 리눅스 설치
나) 소스트리와 makefile을 새로운 리눅스 시스템에 복사하여 어플리케이션을 리빌드
다) 스팍 고유의 코드를 포함하고 있다면, 해당 코드를 수정
라) 포팅된 어플리케이션에 대한 검증 수행
(1) 자바 어플리케이션
(가) 개발툴
(2) 포트란 어플리케이션
(가) GCC 포트란 77 컴파일러, 리눅스용 포트란 90/95 툴
(3) 실행 인터페이스
(4) 시스템 호출(call)과 C 라이브러리
(가) 논리 볼륨, ACL 관리, 로그 기능
(나) 네트워킹
(5) 리눅스에서의 API 호환성 검사
(6) C++ 라이브러리
(7) Math 라이브러리
(8) X 라이브러리와 윈도우 매니저
(가) Xlib와 Xt를 포함하여 X11R6X는 리눅스 상에서 사용 가능
(나) Motif는 오픈 그룹의 오픈 Motif의 형태로 사용 가능
(9) 데스크탑 : CDE와 GNOME/KDE
(10) 쓰레드/LWP(LightWeightProcess) 지원
(11) 프로세스 관리: /proc 파일시스템

5) Performance 튜닝
가) 포팅된 코드의 Performance 확인
나) Performance 분석하는 툴로 Performance Inspector와 OProfile 툴을 이용

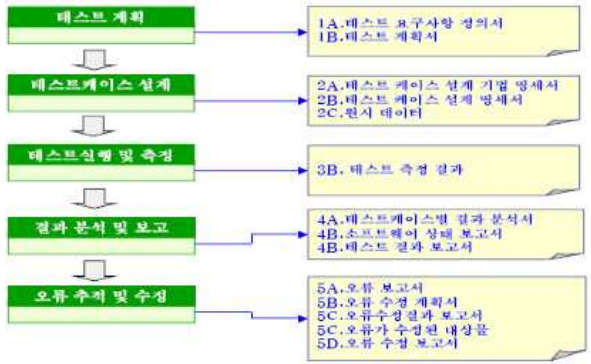
다) Performance Inspector
(1) Performance analysis tool set
(2) 어플리케이션 Performance 문제 및 어플리케이션과 리눅스 커널간의 Performance 수행
라) Profile
(1) RHEL4와 SLES9에 포함되어져 있음
(2) 프로세서 사이클, Translation Look-aside Buffer 실패, 메모리 레퍼런스, 인터럽트 핸들러등의 이벤트 분석

다. 테스트

테스트를 수행함에 있어 우선 테스트 모델링을 고려해 보고 각 단계별로 산출물을 작성할 수 있겠다.



<그림 6 Dual V Model 테스트 모델>



<그림 7 테스트 단계 및 각 단계별 산출물>

라. 유지보수

1) 공개소프트웨어 운영 방안

가) 공개소프트웨어 기반의 시스템 운영

공개소프트웨어 기반 시스템을 운영하기 위해 시스템 구축 업체나 유지보수 업체로부터 아래의 기본 지침을 작성 및 수행하도록 하고 이를 관리한다.

(1) 공개소프트웨어 기반 정보화 시스템 현황

(가) 시스템 전체 구성도
(나) 하드웨어 현황
(다) 소프트웨어 현황
(라) 네트워크 구성도
(마) 서비스 구성도
(바) 보안 구성도

(2) 운영 기준표

시스템의 운영상태 등급을 만들어 시스템 성능 개선 시점의 기준을 설정한다.

(3) 운영지침서

시스템의 표준 운영 지침서를 제작하여 시스템 장애 발생을 최소화한다. 내용은 아래와 같다.

(가) 운영대상 명칭
(나) 운영 서비스의 정의
(다) 설치된 솔루션의 정보 및 버전
(라) 관련 설정 파일 위치 및 설정 내용
(마) 서비스(솔루션) 기동 및 종료 절차
(바) 주변장치 연동 현황
(사) 운영 시 주의사항
(아) 기술지원 담당자 정보

(4) 백업 및 시스템 장애복구 대책

시스템 현황을 기준으로 백업 및 장애복구 관리대상을 선정하고, 예방점검 계획서를 통해 점검을 수행하며 결과를 관리하도록 하며, 백업대상 시스템의 백업 방안 수립하고 수행하도록록 한다.

(5) 보안 지침서

물리적 접근에 대한 보안 정책과 외부로부터의 접근에 대한 보안 정책을 확인하고, 정부에서 인정하는 보안인증 규격(CC, K등급) 준수 사항 등을 확인한다.

2) 공개소프트웨어 유지보수 방안

가) 공개소프트웨어 유지보수 서비스의 필요성

공개소프트웨어는 다른 소프트웨어 패키지와는 달리, 서비스에 대한 책임이며 지속적인 유지보수 계약으로 서비스에 대한 책임이 제공되고 있다.

공개소프트웨어 유지보수는 시스템의 성능과 안정성을 높이고 신기술 적용을 통한 높은 품질의 서비스를 제공할 수 있으며, 서비스에 대한 계약관계이므로 기존 소프트웨어패키지보다 전문적인 서비스를 제공받을 수 있으며 기술지원의 정확성과 신속성이 높다.

나) 공개소프트웨어 유지보수 서비스의 특성과 계약 형태

공개소프트웨어는 라이센스 방식의 소프트웨어패키지와는 제품의 공급 형태가 다르므로 유지보수 서비스 계약에 대한 형태도 다르다. 소프트웨어패키지와 같이 요율제(국내 공공기관의 경우 제품공급가의 8%)로 적용이 되거나 UNIX 시스템의 유지보수 계약과 같이 하드웨어 제조사에서 처리될 경우 그 전문성이 저하되거나 낮은 품질의 서비스가 제공될 우려가 있다. 공개소프트웨어는 초기구입비용이 없거나 현저히 낮은 경우가 일반적이므로 유지보수 서비스 계약 방법 중 요율제의 적용은 어려우며, 정액제로 계약하거나 건별로 서비스를 받는 것이 일반적이다.

다) 공개소프트웨어 유지보수 서비스 내용

(1) 소프트웨어 어플리케이션 및 하드웨어에 대해 인증된 솔루션의 운용 지원 서비스(정합성, 기술 인증 등)
(2) 추가 기능 및 새로운 하드웨어를 제공할 수 있는 정기적인 업데이트를 통한 기능 개선 서비스
(3) 최신 버그 수정 및 보안 패치 서비스
(4) 향후 업그레이드를 비롯하여 모든 릴리즈 버전의 이용 서비스
(5) 신기술 적용을 위한 컨설팅 서비스

5. 결론

이제 리눅스가 경제적이고 유연한 운영체제로 각광을 받으면서 공개SW 리눅스 기반 환경에서 자유로은 IT 인프라를 구축할 수 있게 되었다. 이에 기존 유닉스에서 상용 애플리케이션 환경에 맞는 리눅스 솔루션으로 마이그레이션을 할 때 고려해야 할 요소는 무엇이며 방법론은 무엇인지 알아 보았다.

기존 UNIX 시스템인 Solaris에서 Linux로 마이그레이션을 하는것은 그리 어렵지 않은 사항이다. 왜냐하면 오픈 소스 툴을 사용하여 구축된 애플리케이션이 기존 유닉스 플랫폼에서 정상적으로 작동된다면 리눅스 환경에서도 유사한 툴 및 기능을 제공할 수 있고 대부분의 애플리케이션은 리눅스 환경으로 이동이 가능하기 때문이다. 즉, 리눅스 컴파일러 및 관련 툴은 유닉스 공급업체 툴과 교체할 수 있게 설계되었으며 사용 옵션은 다를 수 있지만 매우 유사한 기능을 제공한다. 특히 애플리케이션 마이그레이션을 위한 가장 중요한 기능은 리눅스가 독점 유닉스 시스템에서 애플리케이션의 이식성을 엄청나게 개선시키는 POSIX와 호환된다는 것이다. 유닉스 애플리케이션의 특징은 멀티 쓰레드 기능인데 최근 상용 리눅스 시스템들도 멀티 쓰레드 기능을 완벽히 구현하도록 설계되어 있어 기존 유닉스 환경에서 리눅스로 쉽게 포팅할 수 있게 되었다. 또한 유닉스 환경의 자바 버츄얼 머신의 경우도 리눅스에서 운영이 가능하기 때문에 자바 애플리케이션을 마이그레이션 하는 것도 그다지 어렵지가 않다. 따라서 일정 기간의 테스팅 후 대부분의 애플리케이션들은 추가 개발이나 수정 없이 마이그레이션을 하면 된다. 이처럼 툴에 대한 모든 문제가 이미 해결되었기 때문에 개발 엔지니어는 운영체제와 프로세서·아키텍처 문제에만 집중하면 된다. 이러한 전략을 통해 신속하고, 예측 가능하며 리스크가 최소화되는 마이그레이션이 가능하다. 고객이 리눅스 플랫폼으로 마이그레이션할 때 얻게 되는 장점은 고객의 다양한 요구사항에 신속하게 대응하게 위해 리눅스 기반에서 개발이 끊임없이 업그레이드 및 신제품이 개발되어 왔다는 점이다.

리눅스는 최적의 운영비용 지출 방법을 강구해야 할 책임이 있는 기업 내 IT예산 관리자와 안정적 기술과 서비스를 제공해야 하는 시스템 관리자들에게 높은 ROI(return on investment,투자 수익)의 창출 기회를 제공하는 최적의 플랫폼이다.

이처럼 사업 또는 기업의 목적 달성을 위해 투자대비 효익을 생각한다면 마이그레이션을 적극적으로 고려해 볼 필요가 있으며 본 문서는 이에 대한 방법론을 제시함으로써 향후 유사 공개SW 마이그레이션 프로젝트를 진행함에 있어 참고자료로 활용을 할 수 있겠다.

6. 참고문헌 및 사이트

마이그레이션_공개SW 마이그레이션가이드, KIPA 2007
리눅스 클라이언트 마이그레이션-구축가이드, KIPA 2007
공개SW유지보수가격정책수립방안, KIPA 2007
테스트기법과방법론, KIPA 2007
http://www.ibm.co.kr
http://www.naver.com