

---

# FreeBSD

---

Korea FreeBSD Users Group – <http://www.kr.FreeBSD.org/>

## Chapter 10 :

<rick@rickinc.com> .

<nobreak@kr.FreeBSD.org> .

: 2004 8 7

: 2005 3 8

: 1.0

FreeBSD Documentation Project FreeBSD Handbook 2004 8

7 . FreeBSD Handbook ,  
, CVS Repository  
,  
,  
가  
가 . KFUG(Korea FreeBSD Users Group)  
KFUG

“FreeBSD Handbook” “FreeBSD Documentation Project” , “ FreeBSD  
” “ (Young-oak Lee)” .

*Copyright © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004 The FreeBSD Documentation Project.*

## 10 장 리눅스 바이너리 호환성

### 10.1 요약

FreeBSD 는 리눅스를 포함하여 여러 다른 유닉스 운영체제와 호환되는 바이너리를 제공한다. 여기서 엄밀하게 왜 FreeBSD 가 리눅스 바이너리를 실행할 수 있어야 하는지? 자문할지도 모른다. 대답은 질문이 아주 단순하다는 것이다. 컴퓨터 세계에서 마지막 이슈가 리눅스기 때문에 수많은 회사와 개발자들은 리눅스를 위해서만 개발한다. 이 문제는 대부분의 이런 회사들이 FreeBSD 버전도 만들었다면 얼마나 많은 사람들이 그들의 제품을 사용할지 모르기 때문에 대부분 리눅스를 위해서만 개발하고 있다. 그래서 FreeBSD 유저는 무엇을 해야 되는가? 이것이 FreeBSD 가 리눅스 바이너리 호환성을 제공하는 이유다.

요약해 보면 이 호환성으로 FreeBSD 유저가 모든 리눅스 어플리케이션의 90%를 수정하지 않고 실행시킬 수 있다. 이들은 **StarOffice™**, 리눅스 버전의 **Netscape®**, **Adobe® Acrobat®**, **RealPlayer® 5** 와 **7**, **VMware™**, **Oracle®**, **WordPerfect®**, **Doom**, **Quacke** 등 다수의 어플리케이션을 포함한다. 특정 상황에서는 리눅스 보다 FreeBSD 에서 리눅스 바이너리 성능이 더 좋다고 보고된 것도 있다.

그러나 FreeBSD 에서 지원하지 않는 특정 리눅스 운영체제에서만 실행되는 것도 있다. 리눅스 /proc 파일 시스템이나 가상 8086 모드(8086 의 확장된 개념으로 8086 에서 만들어진 프로그램과 호환될 수 있도록 하는 방식) 활성화처럼 i386™의 특정 콜을 과도하게 사용하는 리눅스 바이너리는 FreeBSD 에서 동작하지 않는다.

이번 장을 읽고 다음과 같은 사항을 알 수 있다:

- 시스템에서 리눅스 바이너리 호환성을 어떻게 활성화하는가.
- 추가적인 리눅스 공유 라이브러리는 어떻게 설치하는가.
- FreeBSD 시스템에 리눅스 어플리케이션을 어떻게 설치하는가.
- FreeBSD의 리눅스 호환성에 대한 자세한 사항

이번 장을 읽기 전에 다음 사항을 알고 있어야 한다:

- 소프트웨어를 어떻게 설치하는가 (4장)

## 10.2 설치

리눅스 바이너리 호환성은 기본적으로 꺼져 있다. 이 기능을 사용하는 가장 쉬운 방법은 *linux* KLD 오브젝트를("커널이 로드할 수 있는 오브젝트") 로드한다. 명령 프롬프트에서 *linux* 만 입력하여 이 모듈을 로드할 수 있다. 설치하는 동안 리눅스 바이너리 호환성을 설치했다면 다음과 같은 결과를 볼 수 있다:

```
# linux
Linux driver already loaded
```

리눅스 호환성을 항상 사용하려면 다음 라인을 */etc/rc.conf* 에 추가한다:

```
linux_enable="YES"
```

*kldstat(8)* 명령은 KLD 가 로드 되었는지 확인하는데 사용할 수 있다:

% kldstat				
Id	Refs	Address	Size	Name
1	2	0xc0100000	16bdb8	kernel
7	1	0xc24db000	d000	linux.ko

어떤 이유로 KLD 로드를 원치 않거나 로드할 수 없다면 커널 설정파일에 *options COMPAT\_LINUX*를 추가하여 리눅스 바이너리 호환성을 커널에 정적으로 링크한다. 8 장의 설명대로 새로운 커널을 설치한다.

### 10.2.1 리눅스 런타임 라이브러리 설치

두 가지 방법 중 한가지를 사용하여 설치할 수 있다; *linux\_base* 포트를 사용하거나 이들을 수동으로 설치하면 된다.

### 10.2.1.1 linux\_base 포트를 사용하여 설치하기

이것이 런타임 라이브러리를 설치하는 가장 쉬운 방법이다. 단순히 포트 컬렉션에서 다른 포트처럼 설치하면 된다. 다음 명령을 실행한다:

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

이제 리눅스 바이너리 호환성이 작동한다. 어떤 프로그램은 정확하지 않은 시스템 라이브러리 버전으로 문제가 발생한다. 그러나 이것은 보통 문제가 되지 않는다.

**Note:** 다른 버전의 리눅스 배포본처럼 여러 가지 버전의 emulators/linux\_base 포트가 있을 것이다. 설치를 원하는 리눅스 어플리케이션에서 필요한 가장 최신 포트를 설치하면 된다.

### 10.2.1.2 수동으로 라이브러리 설치

"포트" 컬렉션이 설치되지 않았다면 수동으로 라이브러리를 설치할 수 있다. 런타임 링커(linker)에 의존하는 프로그램을 사용할 때 리눅스 공유 라이브러리가 필요하다. 또한 FreeBSD 시스템의 리눅스 라이브러리를 위해 "shadow root" 디렉터리 /compat/linux 를 생성해야 된다. FreeBSD 에서 실행되는 모든 리눅스 프로그램은 공유 라이브러리를 열고 이 트리를 처음으로 확인한다. 그래서 예를 들면 리눅스 프로그램이 /lib/libc.so 를 로드한다면 FreeBSD 는 /compat/linux/lib/libc.so 를 처음으로 열고 이 파일이 없다면 /lib/libc.so 를 연다. 공유 라이브러리는 리눅스 ld.so 리포트의 경로보다 shadow 트리 /compat/linux/lib 에 설치되어야 한다.

FreeBSD 시스템에 설치한 리눅스 바이너리가 처음에만 의존하는 공유 라이브러리를 찾아야 한다. 잠시 후 추가적인 작업 없이 새롭게 설치된 리눅스 바이너리를 실행할 수 있는 만족할 만한 리눅스 공유 라이브러리를 시스템에 설치할 수 있다.

### 10.2.1.3 부가적인 공유 라이브러리를 어떻게 추가하는가

linux\_base 포트를 설치했지만 아직도 어플리케이션이 다른 공유 라이브러리를 원한다면 어떤 공유 라이브러리를 리눅스 바이너리가 필요로 하고 어디서 구할 수 있는지 어떻게 알

수 있는가? 기본적으로 2 가지 가능성이 있다(FreeBSD 시스템에서 다음 지시 사항을 따를 때 root 권한이 필요하다).

리눅스 시스템을 사용할 수 있다면 어떤 공유 라이브러리가 어플리케이션에 필요한지 살펴보고 FreeBSD 시스템에 복사한다. 다음 예제를 확인한다:

리눅스 바이너리의 **Doom** 을 FTP 로 받아왔고 이것을 리눅스 시스템에 넣었다고 가정한다. 어떤 공유 라이브러리가 필요한지 ldd linuxdoom 을 실행해서 다음과 같이 체크할 수 있다:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

마지막 칼럼까지 모든 파일을 가져와서 심볼릭 링크가 지정하듯이 /compat/linux 밑에 넣는다. 이 말은 드디어 FreeBSD 시스템에 이들 파일을 가질 수 있다는 의미다:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Note:** 이미 ldd 출력의 첫 번째 칼럼과 똑같은 버전의 리눅스 공유 라이브러리를 가지고 있다면 공유 라이브러리가 동작하고 있기 때문에 마지막 칼럼에 있는 이름의 파일을 시스템에 복사할 필요가 없다. 새로운 버전의 공유 라이브러리면 어쨌든 복사하는 것을 권장한다. 새로운 것을 심볼릭 링크로 만들었다면 예전 것을 삭제할 수 있다.

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

그래서 이러한 공유 라이브러리를 시스템에 가지고 있다면 ldd 의 출력에 따라 새로운 버전의 필요한 바이너리를 찾는다:

```
libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
```

마지막 한 두 개의 버전만 다르다면 약간 예전 버전의 프로그램에서 정확히 동작하기 때문에 /lib/libc.so.4.6.29 를 복사하면 된다. 그러나 원한다면 libc.so 로 교체할 수 있다.

```
/compat/linux/lib/libc.so.4.6.29  
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Note:** 심볼릭 링크 메커니즘은 리눅스 바이너리에만 필요하다. FreeBSD 런타임 링커는 주 버전 번호가 매칭되는지 확인하기 때문에 걱정할 필요는 없다.

## 10.2.2 리눅스 ELF 바이너리 설치

ELF 바이너리는 가끔 추가적인 "브랜딩(branding)" 단계가 필요하다. 브랜드가 없는 ELF 바이너리를 실행하려고 하면 다음과 같은 에러 메시지를 보게 된다:

```
% ./my-linux-elf-binary  
ELF binary type not known  
Abort
```

리눅스 바이너리에서 FreeBSD ELF 바이너리를 FreeBSD 커널이 구분할 수 있도록 brandelf(1) 유틸리티를 사용한다.

```
% brandelf -t Linux my-linux-elf-binary
```

이제 GNU 툴 체인이(toolchain) 적당한 브랜딩 정보를 ELF 바이너리에 자동으로 넣기 때문에 나중에 이 단계가 필요없다.

## 10.2.3 호스트 네임 리졸버(resolver) 설정

DNS 가 작동하지 않거나 다음과 같은 메시지를 보게 된다면:

```
resolv+: "bind" is an invalid keyword resolv+:
```

"hosts" is an invalid keyword

/compat/linux/etc/host.conf 파일에 다음 내용을 추가해야 된다:

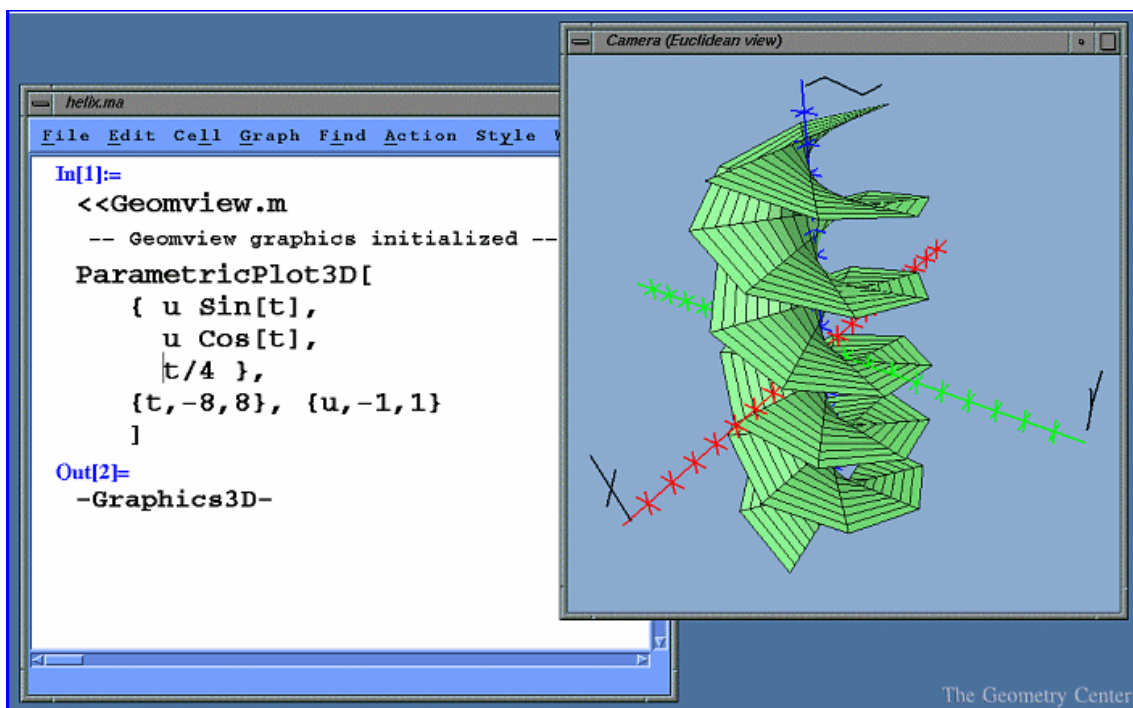
```
order hosts, bind
multi on
```

여기서 순서는 처음에 /etc/hosts 를 검색하고 DNS 는 두 번째로 검색하도록 지정한다. /compat/linux/etc/host.conf 가 설치되지 않았을 때 리눅스 어플리케이션은 FreeBSD 의 /etc/host.conf 를 찾고 맞지 않는 FreeBSD 구문에 대해 알려준다. /etc/resolv.conf 파일로 네임서버를 설정하지 않았다면 *bind(DNS 서버)*를 삭제한다.

## 10.3 Mathematica® 설치

이 문서는 리눅스 버전의 Mathematica® (4.X)를 FreeBSD 시스템에 설치하는 단계를 설명한다.

Mathematica 샘플 예제



[10-1.bmp]

리눅스 버전의 **Mathematica** 는 FreeBSD 에서 완벽하게 동작하지만 바이너리에 Wolfram 브랜드 적용이 필요하기 때문에 FreeBSD 는 리눅스 ABI 로 이 바이너리를 실행하는 것을 알고 있다.

리눅스 버전의 **Mathematica** 나 학생용 **Mathematica** 는 직접 Wolfram <http://www.wolfram.com/>에서 구입할 수 있다.

### 10.3.1 리눅스 바이너리 브랜딩

리눅스 바이너리는 Wolfram 이 배포하는 **Mathematica** CDROM 의 Unix 디렉터리에 있다. 이 디렉터리 트리를 로컬 하드 드라이브로 복사해서 인스톨러를 실행하기 전에 `brandelf(1)`으로 리눅스 바이너리를 브랜드 할수 있다:

```
# mount /cdrom
# cp -rp /cdrom/Unix/ /localdir/
# brandelf -t Linux /localdir/Files/SystemFiles/Kernel/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/FrontEnd/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Installation/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Graphics/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Converters/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/LicenseManager/Binaries/Linux/mathlm
# cd /localdir/Installers/Linux/
# ./MathInstaller
```

다른 방법은 다음 명령으로 브랜드되지 않은 모든 바이너리에 기본 ELF 브랜드를 설정할 수 있다:

```
# sysctl kern.fallback_elf_brand=3
```

위 명령은 FreeBSD 가 브랜드 되지 않은 ELF 바이너리에 리눅스 ABI 를 사용하도록 하기 때문에 CDROM 에서 직접 인스톨러를 실행할 수 있다.



## 10.3.2 Mathematica 패스워드 받기

**Mathematica** 를 실행하기 전에 머신 ID 에 맞는 패스워드를 Wolfram 에서 받아야 한다.

리눅스 호환성 런타임 라이브러리를 설치한 후 설치 디렉터리의 압축이 풀린

**Mathematica** 에서 mathinfo 프로그램을 실행하여 머신 ID 를 받는다. 이 머신 ID 는 첫 번째 이더넷 카드의 MAC 주소와 관련 있다.

```
# cd /localdir/Files/SystemFiles/Installation/Binaries/Linux
# mathinfo
disco.example.com 7115-70839-20412
```

메일, 전화 또는 팩스로 Wolfram 에 등록할때 머신 ID 를 그들에게 주면 그룹으로 이루어진 그룹 패스워드를 받는다. 플랫폼에 상관없이 처음으로 **Mathematica** 를 실행할 때 이 정보를 입력한다.

## 10.3.3 네트워크로 Mathematica 를 전면에서 운용하기

**Mathematica** 는 표준 폰트셋으로(정수, 합, 그리스 문자 등) 표현하지 못하는 문자를 표시하기 위해 특수한 폰트를 사용한다. X 프로토콜은 이들 문자가 로컬에 설치되어 있어야 한다. 이 의미는 이들 폰트를 CDROM 또는 **Mathematica** 가 설치된 호스트로부터 로컬 머신에 복사해야 된다는 것이다. 이들 폰트는 보통 CDROM 의 /cdrom/Unix/Files/SystemFiles/Fonts 에 있거나 하드 드라이브의 /usr/local/mathematica/SystemFiles/Fonts 에 있다. 실재 폰트는 Type1 과 X 서브 디렉터리에 있다. 위에서 설명했듯이 이들을 사용하는 여러 가지 방법이 있다.

첫 번째 방법은 /usr/X11R6/lib/X11/fonts 의 폰트디렉터리 한곳에 복사하는 것이다. 이 방법은 fonts.dir 파일에 폰트이름을 넣고 첫 번째 라인의 수많은 폰트번호를 바꾸어야 한다. 그렇지 않으면 폰트를 복사해 넣은 디렉터리에서 mkfontdir(1)를 실행하면 된다.

두 번째 방법은 디렉터리를 /usr/X11R6/lib/X11/fonts 에 복사하는 것이다:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
```

```
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

이제 새로운 폰트디렉터리를 폰트경로에 추가한다:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

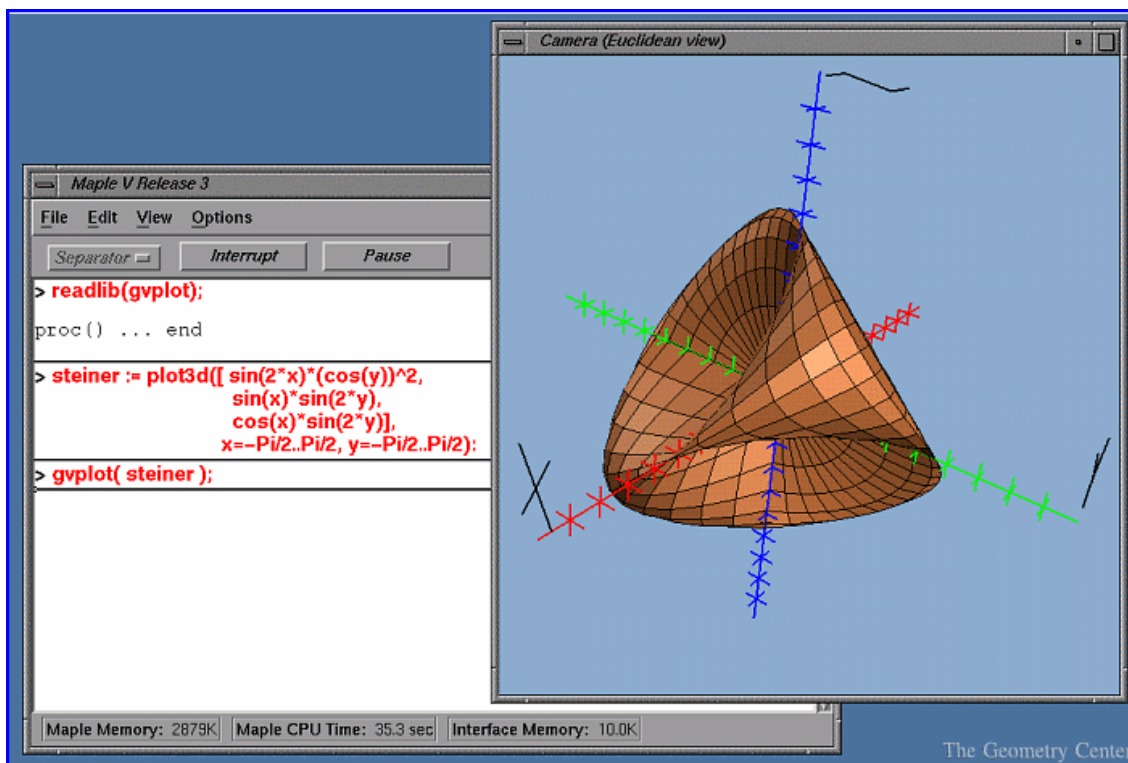
XFree86™ 서버를 사용 중이라면 XF86Config 파일에 이들 폰트 디렉터리를 추가하여 자동으로 로드할 수 있다.

/usr/X11R6/lib/X11/fonts/Type1 이라는 디렉터리를 가지고 있지 않다면 위의 예제에서 MathType1 의 이름을 Type1 로 바꿀 수 있다.

## 10.4 Maple™ 설치

Maple™ 은 Mathematica®와 비슷한 상용 계산프로그램이다. 이 소프트웨어를 <http://www.maplesoft.com/>에서 구입하고 라이선스 파일을 등록해야 된다.

Maple 샘플 예제



[10-2.bmp]

이 소프트웨어를 FreeBSD 에 설치하려면 단순히 다음 단계를 따른다.

**[따라 하기 예제: Maple 설치]**

- ① 제품에서 INSTALL 셸 스크립트를 실행한다. 설치 프로그램에 프롬프트가 나올 때 "RedHat" 옵션을 선택한다. 일반적인 설치 디렉터리는 /usr/local/maple이다.
  - ② 설치가 끝나지 않는다면 Maple Waterloo Software (<http://register.maplesoft.com>)에서 라이선스를 신청해서 이 라이선스를 /usr/local/mple/license/license.dat에 복사 한다.
  - ③ Maple에있는 INSTALL\_LIC 설치 스크립트를 실행하여 FLEXlm 라이선스를 설치한다. 라이선스 서버에게 머신의 주 호스트 이름을 지정한다.
  - ④ 다음 내용으로 /usr/local/mple/bin/mple.system.type 파일을 패치한다:
- snip -----

```

*** maple.system.type.orig      Sun Jul  8 16:35:33 2001
--- maple.system.type      Sun Jul  8 16:35:51 2001
*****
*** 72,77 ****
--- 72,78 ----
        # the IBM RS/6000 AIX case
        MAPLE_BIN="bin.IBM_RISC_UNIX"
        ;;
+   "FreeBSD"|W
    "Linux")
        # the Linux/x86 case
        # We have two Linux implementations, one for Red Hat and
----- snip end of patch -----

```

"FreeBSD"|W 뒤에 공백이 들어가지 않도록 주의한다.

이 패치는 **Maple** 이 "FreeBSD"를 리눅스 시스템의 일종으로 감지하도록 한다.

bin/maple 쉘 스크립트는 운영체제 이름을 찾기 위해 `uname -a` 를 수행하는 `bin/maple.system.type` 쉘 스크립트를 불러온다. OS 이름에 따라 어떤 바이너리를 사용할지 찾는다.

#### ⑤ 라이선스 서버 시작

/usr/local/etc/rc.d/lmgrd.sh 에 설치되어있는 다음 스크립트가 lmgrd 를 시작하는 편리한 방법이다:

```

----- snip -----

#!/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

```

```
case "$1" in
start)
    Imgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
    echo -n " Imgrd"
    ;;
stop)
    Imgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
    ;;
*)
    echo "Usage: `basename $0` {start|stop}" 1>&2
    exit 64
    ;;
esac

exit 0

----- snip -----
```

⑥ **Maple** 테스트 시작:

```
% cd /usr/local/maple/bin
% ./xmaple
```

Maplesoft 에게 여러분은 FreeBSD 버전의 프로그램이 필요하다고 메일을 보낸다.

[따라 하기 예제 끝]

## 10.4.1 일반적인 함정

- **FLEXIm** 라이선스 매니저는 동작시키기 어려운 툴이다. 추가적인 문서는 <http://www.globetrotter.com/>에서 찾을 수 있다.
- 문제가 발생했다면 Imgrd는 라이선스 파일과 core dump에 대해 까다로운 것으로

알려져 있다. 정확한 라이선스 파일은 다음과 같이 생겼다:

```
# =====  
# License File for UNIX Installations ("Pointer File")  
# =====  
SERVER chillig ANY  
#USE_SERVER  
VENDOR mapleimg  
  
FEATURE Maple mapleimg 2000.0831 permanent 1 XXXXXXXXXXXX W  
    PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." W  
    ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" W  
    SN=XXXXXXXXXX
```

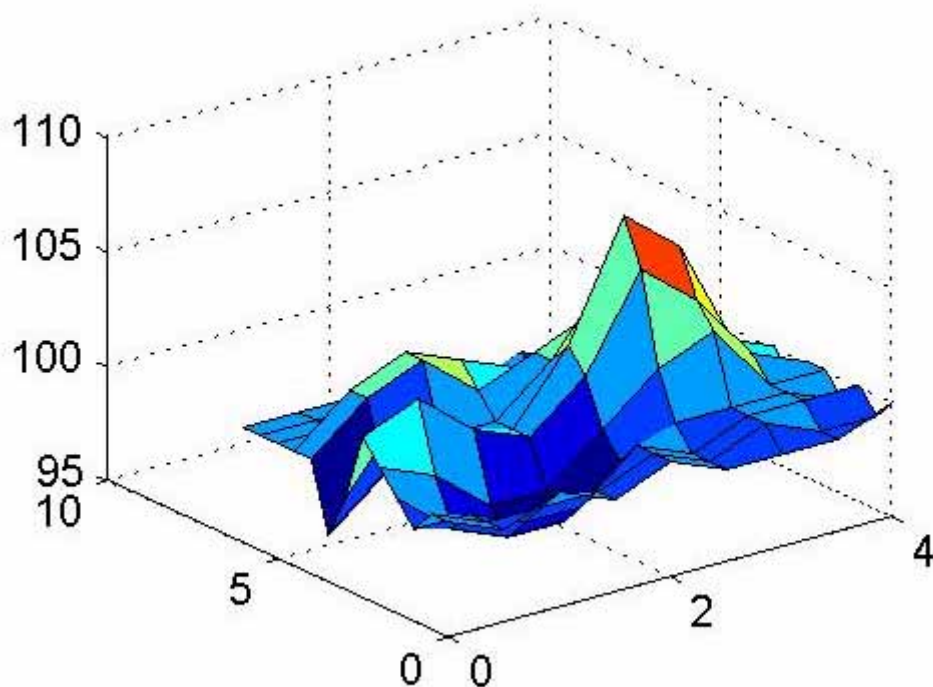
**Note:** 시리얼 번호와 키는 "X"로 나타냈다. chillig 는 호스트 네임이다.

"FEATURE" 라인을(라이선스 키로 보호되기 때문에) 건드리지 않고 라이선스 파일을 수정한다.

## 10.5 MATLAB® 설치

이 문서는 리눅스 버전의 **MATLAB®** 버전 6.5 를 FreeBSD 시스템에 설치하는 절차에 대해 설명한다. **Java** 가상 머신을(10.5.3 장을 본다) 제외하고 정확히 동작한다.

**MATLAB 샘플 예제**



[10-3.bmp]

리눅스 버전의 **MATLAB** 는 MathWorks <http://www.mathworks.com> 에서 직접 구입할 수 있다. 또한 라이선스 파일을 받거나 생성하는 방법도 알수 있다. 그들에게 여러분은 FreeBSD 버전의 소프트웨어가 필요하다고 얘기한다.

## 10.5.1 MATLAB 설치

MATLAB 설치는 다음 절차를 따른다:

- ① 설치 스크립트에서 권장하듯이 root에서 설치 CD를 넣고 마운트한다. 설치 스크립트를 시작하려면 다음 명령을 입력한다:

```
# /compat/linux/bin/sh /cdrom/install
```

**Tip:** 인스톨러는 그래픽으로 보여준다. 디스플레이를 열수 없다는 에러 메시지를 보여 주면 `setenv HOME ~ USER`를 입력한다. *USER*는 `su(1)`를 수행하기 전의 유저다.

- ② **MATLAB** root 디렉터리를 요청하면 `/compat/linux/usr/local/matlab`을 입력한다:

**Tip:** 나머지 설치 절차에서 입력하기 쉽게 셸 프롬프트에 입력한다: `set MATLAB=/compat/linux/usr/local/matlab`

- ③ **MATLAB** 라이선스를 받았을 때 설명한 대로 라이선스 파일을 수정한다.

**Tip:** 인스톨러가 라이선스 파일을 수정하라고 요청하기 전에 좋아하는 에디터로 이 파일을 준비해서 `$MATLAB/license.dat`에 복사한다.

- ④ 설치 절차를 끝낸다.

이제 **MATLAB** 설치가 끝났다. 다음 단계는 FreeBSD 시스템에 "glue"를 적용한다.

## 10.5.2 라이선스 매니저 시작

- ① 라이선스 매니저 스크립트에 심볼릭 링크를 생성한다:

```
# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
```

- ② `/usr/local/etc/rc.d/flexlm.sh`에 시작 파일을 생성한다. 아래 예제는 배포된 `$MATLAB/etc/rc.lm.glnx86`의 수정된 버전이다. 파일 위치를 변경하고 리눅스 에뮬레이션에서 라이선스 매니저를 시작한다.

```
#!/bin/sh
case "$1" in
start)
    if [ -f /usr/local/etc/lmboot_TMW ]; then
        /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u
```



```
username && echo 'MATLAB_lmgrd'
    fi
    ;;
stop)
    if [ -f /usr/local/etc/lmdown_TMW ]; then
        /compat/linux/bin/sh /usr/local/etc/lmdown_TMW > /dev/null
2>&1
    fi
    ;;
*)
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac

exit 0
```

**중요:** 파일을 실행할 수 있어야 한다:

```
# chmod +x /usr/local/etc/rc.d/flexlm.sh
```

위의 *username* 은 시스템에 있는 유저 이름을(root 가 아닌) 넣는다.

③ 다음 명령으로 라이선스 매니저를 시작한다:

```
# /usr/local/etc/rc.d/flexlm.sh start
```

### 10.5.3 Java 런타임 환경 링크

FreeBSD 에서 동작하도록 Java 런타임 환경의 링크를 변경한다:

```
# cd $MATLAB/sys/java/jre/glnx86/
# unlink jre; ln -s ./jre1.1.8 ./jre
```

## 10.5.4 MATLAB 시작 스크립트 생성

- ① 다음 시작 스크립트를 /usr/local/bin/matlab에 입력한다:

```
#!/bin/sh
/compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
```

- ② 그리고 `chmod +x /usr/local/bin/matlab` 명령을 실행한다.

**Tip:** `emulators/linux_base` 버전에 따라 이 스크립트를 실행할 때 에러를 볼 수 있다. 에러를 피하기 위해 `/compat/linux/usr/local/matlab/bin/matlab` 파일을 다음과 같이 변경한다:

```
if [ `expr "$lscmd" :'.*->.*' -ne 0` ];
```

위의 라인을 (버전 13.0.1에서는 410 라인이다.) 다음 라인으로 변경한다:

```
if test -L $newbase; then
```

## 10.5.5 MATLAB 섯다운 스크립트 생성

다음은 MATLAB 가 정확히 끝나지 않는 문제를 해결하기 위해 필요하다.

- ① `$MATLAB/toolbox/local/finish.m` 파일을 만들고 다음 라인을 입력한다:

```
! $MATLAB/bin/finish.sh
```

**Note:** `$MATLAB` 는 문자다.

**Tip:** 빠져 나가기 전의 작업 환경을 저장하는 `finis sav.m` 과 `finishdig.m` 파일을 같은 디렉터리에서 찾을 수 있다. 이들 중 하나를 사용한다면 `save` 명령을 입력하고 즉시 위 라인을 입력한다.

- ② 다음 내용이 있는 `$MATLAB/bin/finish.sh` 파일을 생성한다:

```
#!/usr/compat/linux/bin/sh
(sleep 5; killall -1 matlab_helper) &
exit 0
```

- ③ 파일을 실행할 수 있게 만든다:

```
# chmod +x $MATLAB/bin/finish.sh
```

## 10.5.6 MATLAB 사용

matlab 를 입력해서 사용할 준비가 됐다.

## 10.6 오라클 설치

### 10.6.1 서문

이 문서는 리눅스용 오라클 8.0.5 와 오라클 8.0.5.1 엔터프라이즈 에디션을 FreeBSD 머신에 설치하는 과정을 설명한다.

### 10.6.2 리눅스 환경 설치

emulators/linux\_base 와 devel/linux\_devtools를 포트 컬렉션에서 설치해야 된다. 이들을 포트에서 설치하기 어렵다면 패키지나 포트 컬렉션에서 예전 버전을 사용할 수 있다.

인텔리전트 에이전트를 사용하겠다면 Red Hat Tcl 패키지도 설치해야 된다: tcl-8.0.3-20.i386.rpm. 공식 RPM 포트에서(archivers/rpm) 패키지를 설치하는 일반적인 명령은 다음과 같다:

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm package
```

패키지 설치에 어려움이 없어야 한다.

## 10.6.3 오라클 환경 생성

오라클을 설치하기 전에 적당한 환경을 설정해야 된다. 이 문서는 오라클 설치 가이드가 아닌 리눅스용 오라클을 FreeBSD 에 설치할때 특별히 필요한 것에 대해 설명한다.

### 10.6.3.1 커널 튜닝

오라클 설치 가이드에서 설명하듯이 최대 크기의 공유 메모리를 설정해야 된다. FreeBSD 에서는 *SHMMAX* 를 사용하지 않는다. *SHMMAX* 는 단지 *SHMMAXPGS* 와 *PGSIZE* 를 계산할 뿐이다. 따라서 *SHMMAXPGS* 를 정의해야 된다. 여기에 사용할 수 있는 다른 모든 옵션은 가이드에 설명되어 있다. 예를 들면 다음과 같이 정의한다:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

이들 옵션을 사용하려는 오라클 용도에 맞게 설정한다.

그리고 다음 옵션을 커널 설정 파일에 지정한다:

```
options SYSVSHM #SysV shared memory
options SYSVSEM #SysV semaphores
options SYSVMSG #SysV interprocess communication
```

### 10.6.3.2 오라클 계정

다른 계정을 만들듯이 *oralc* 계정을 만든다. *oracle* 계정은 특별하기 때문에 리눅스 셸을 줘야 된다. */etc/shells* 에 */compat/linux/bin/bash* 를 추가하고 *oracle* 계정 셸을 */compat/linux/bin/bash* 로 설정한다.

### 10.6.3.3 환경

ORACLE\_HOME 과 ORACLE\_SID 같은 일반적인 오라클 환경 외에도 다음 환경 변수를 지정해야 된다:

변수	값
LD_LIBRARY_PATH	<i>\$ORACLE_HOME/lib</i>
CLASSPATH	<i>\$ORACLE_HOME/jdbc/lib/classes111.zip</i>
PATH	<i>/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin \$ORACLE_HOME/bin</i>

모든 환경 변수를 .profile 에 설정할 것을 권장한다. 전체적인 예제는 다음과 같다:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

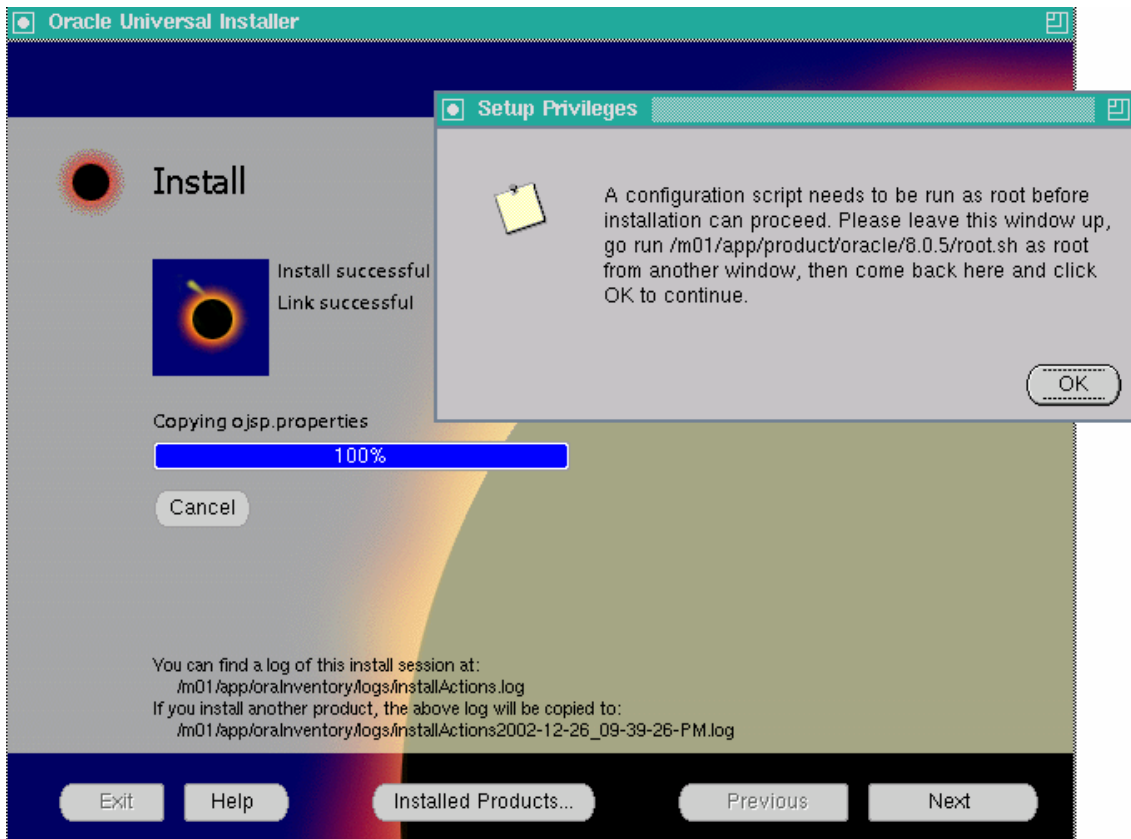
### 10.6.4 오라클 설치

리눅스 에뮬레이터의 약간의 불일치로 인스톨러를 시작하기 전에 /var/tmp 에 .oracle 이라는 이름의 디렉토리를 생성해야 된다. 누구나 쓰기를 할 수 있거나 oracle 유저만 쓰기를 하도록 한다. 오라클을 아무런 문제없이 설치할 수 있다. 문제가 발생 한다면 첫째로 오라클 배포판이나 설정을 체크한다. 오라클을 설치하고 다음 두 개의 서브 섹션에서 설명하는 패치를 적용한다.

가끔 발생하는 문제는 TCP 프로토콜 아답터가 제대로 설치되지 않는 것이다. 그 결과 어떤 TCP 리스너(listener)도 시작하지 못한다. 다음은 이 문제를 해결하는데 도움을 준다:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

root.sh 를 다시 실행하는 것을 잊지 않는다!



[10-4.bmp]

### 10.6.4.1 root.sh 패치

오라클을 설치할 때 root 에서 실행해야 되는 몇 가지가 root.sh 라는 스크립트에 저장되어 있다. 이 스크립트는 orainst 디렉터리에 작성되어 있다. 다음 패치를 root.sh 에 적용하고 chown 를 사용할 적당한 위치로 이동시키거나 리눅스 셸에서 스크립트를 실행한다.

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
*****
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

오라클을 CD 에서 설치하지 않았다면 root.sh 소스를 패치할 수 있다. 이것은 rthd.sh 라고 하고 소스 트리의 orainst 디렉터리에 있다.

### 10.6.4.2 genclntsh 패치

스크립트 genclntsh 는 싱글 공유 클라이언트 라이브러리를 생성하는데 사용된다. demo 를 빌드할 때 사용된다. PATH 주석을 풀어서 다음 패치를 적용한다:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
*****
*** 32,38 ****
#
```

```
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

## 10.6.5 오라클 실행

이 지시를 따랐다면 리눅스에서 실행하는 것처럼 **오라클**을 실행할 수 있다.

## 10.7 SAP® R/3® 설치

SAP® 시스템을 FreeBSD 에 설치하는 것은 SAP 지원 팀이 지원하지 않는다.

### 10.7.1 서문

이 문서는 FreeBSD 와 **오라클** 설치를 포함하여 FreeBSD 머신에 **SAP R/3** 시스템과 리눅스용 **오라클** 데이터베이스를 설치할 수 있는 방법을 설명한다. 두 가지 설정이 설명된다:

- FreeBSD 4.3-STABLE에 **오라클 8.0.5**와 **SAP R/3 4.6B (IDES)**
- FreeBSD 4.5-STABLE에 **오라클 8.1.7**과 **SAP R/3 4.6C**

이 문서가 중요한 모든 단계를 아주 자세히 설명하더라도 **오라클**과 **SAP R/3** 설치 가이드를 대체할 수 없다.



오라클과 SAP OSS 리소스뿐만 아니라 SAP 는 SAP R/3 리눅스용 에디션의 문서와 오라클 관련 질문도 참고한다.

## 10.7.2 소프트웨어

다음 CD-ROM 들이 SAP 설치에 사용된다:

### 10.7.2.1 SAP R/3 4.6B, 오라클 8.0.5

이름	번호	설명
KERNEL	51009113	SAP Kernel Oracle / Installation / AIX, Linux, Solaris
RDBMS	51007558	Oracle / RDBMS 8.0.5.X / Linux
EXPORT1	51010208	IDES / DB-Export / Disc 1 of 6
EXPORT2	51010209	IDES / DB-Export / Disc 2 of 6
EXPORT3	51010210	IDES / DB-Export / Disc 3 of 6
EXPORT4	51010211	IDES / DB-Export / Disc 4 of 6
EXPORT5	51010212	IDES / DB-Export / Disc 5 of 6
EXPORT6	51010213	IDES / DB-Export / Disc 6 of 6

추가적으로 사실 필요하지 않은 오라클 8 서버 CD 와(리눅스용 버전 8.0.5, 커널 버전 2.0.33) FreeBSD 4.3-STABLE 를 사용했다(4.3 릴리즈 보다 며칠 전의).

### 10.7.2.2 SAP R/3 4.6C SR2, 오라클 8.1.7

이름	번호	설명
KERNEL	51014004	SAP Kernel Oracle / SAP Kernel Version 4.6D / DEC, Linux
RDBMS	51012930	Oracle 8.1.7/ RDBMS / Linux
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 1 of 4

EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 2 of 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 3 of 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Disc 4 of 4
LANG1	51013954	Release 4.6C SR2 / Language / DE, EN, FR / Disc 1 of 3

설치 하려는 언어에 따라 언어 CD 가 추가적으로 필요할 것이다. 우리는 DE 와 EN 을 사용하기 때문에 첫 번째 언어 CD 만 필요하다. 4 개의 EXPORT CD 번호는 모두 동일하다. 또한 3 개의 언어 CD 도 같은 번호를 가지고 있다 (이것은 4.6B IDES 릴리즈 CD 부터 달라졌다). 이 문서를 작성할 때 FreeBSD 4.5-STABLE 에서(20.03.2002) 설치했다.

### 10.7.3 SAP 노트

다음 노트를 SAP R/3 를 설치하기 전에 읽으면 설치하는 동안 유용하다:

#### 10.7.3.1 SAP R/3 4.6B, 오라클 8.0.5

번호	제목
0171356	SAP Software on Linux: Essential Comments
0201147	INST: 4.6C R/3 Inst. on UNIX - Oracle
0373203	Update / Migration Oracle 8.0.5 --> 8.0.6/8.1.6 LINUX
0072984	Release of Digital UNIX 4.0B for Oracle
0130581	R3SETUP step DIPGNTAB terminates
0144978	Your system has not been installed correctly
0162266	Questions and tips for R3SETUP on Windows NT / W2K

#### 10.7.3.2 SAP R/3 4.6C 오라클 8.1.7

번호	제목
0015023	Initializing table TCPDB (RSXP0004) (EBCDIC)
0045619	R/3 with several languages or typefaces

0171356	SAP Software on Linux: Essential Comments
0195603	RedHat 6.1 Enterprise version: Known problems
0212876	The new archiving tool SAPCAR
0300900	Linux: Released DELL Hardware
0377187	RedHat 6.2: important remarks
0387074	INST: R/3 4.6C SR2 Installation on UNIX
0387077	INST: R/3 4.6C SR2 Inst. on UNIX – Oracle
0387078	SAP Software on UNIX: OS Dependencies 4.6C SR2

## 10.7.4 필요한 하드웨어

다음 장비는 **SAP R/3 시스템** 설치에 적당하다. 제품을 사용하기 위해 더 많은 공간이 필요하다:

컴포넌트	4.6B	4.6C
Processor	2 x 800MHz Pentium® III	2 x 800MHz Pentium III
Memory	1GB ECC	2GB ECC
Hard Disk Space	50-60GB (IDES)	50-60GB (IDES)

제품을 사용하기 위해 대형 케이스의 Xeon 프로세서와 고속 디스크 접근(SCSI, RAID 하드웨어 컨트롤러) 그리고 USV 와 ECC-RAM 이 권장된다. 설치하는 동안 27GB 의 데이터베이스 파일을 생성하는 거대한 양의 하드 디스크 공간이 IDES 시스템에 이미 설정되어 있다. 또한 이 공간은 제품 시스템과 어플리케이션 데이터를 초기화하는데 충분하다.

### 10.7.4.1 SAP R/3 4.6B, 오라클 8.0.5

다음은 규격 하드웨어를 사용했다: 2 개의 800 MHz 펜티엄 III 프로세서가 있는 듀얼 프로세서 보드, Adaptec 29160 울트라 160 SCSI 아답터(40/80 GB DLT 테잎 드라이브와 CDROM 을 사용하기 위해), Mylex AcceleRAID (2 채널, 펌웨어 6.00-1-00 과 32MB 램). Mylex RAID 컨트롤러를 두 개의 17GB 하드 디스크와(미러 된) 4 개의 36GB 하드 디스크에(RAID 레벨 5) 연결한다.

## 10.7.4.2 SAP R/3 4.6C, 오라클 8.1.7

여기서는 Dell PowerEdge 2500 이 사용되었다: 두 개의 1G 펜티엄 III 프로세서의(256 kB 캐시) 듀얼 프로세서 보드, 2GB PC 133ECC SDRAM, 128MB 의 PERC/3 DC PCI RAID 컨트롤러 그리고 EIDE DVD-ROM 드라이브. RAID 컨트롤러에 두 개의 18GB 하드 디스크와(미러 된) 4 개의 36GB 하드 디스크를(RAID 레벨 5) 붙인다.

## 10.7.5 FreeBSD 설치

첫째 FreeBSD 를 설치해야 된다. FreeBSD 를 설치하는 여러 가지 방법이 있다(FreeBSD 4.3 은 FTP 를 통해 그리고 FreeBSD 4.5 는 릴리즈 CD 에서 직접 설치하였다). 더 많은 정보는 2.13 장을 본다.

### 10.7.5.1 디스크 레이아웃

단순히 같은 디스크 레이아웃을 SAP R/3 46B 와 SAP R/3 46C SR2 설치에 사용했다. 다른 하드웨어에(/dev/da 와 /dev/amr 을 각각 사용했기 때문에 AMI MegaRAID 를 사용한다면 /dev/da0s1a 대신 /dev/amr0s1a 를 볼수 있다) 설치했기 때문에 장치 이름만 변경했다:

파일 시스템	크기(1k-블록)	크기(GB)	마운트 포인트
/dev/da0s1a	1.016.303	1	/
/dev/da0s1b		6	swap
/dev/da0s1e	2.032.623	2	/var
/dev/da0s1f	8.205.339	8	/usr
/dev/da1s1e	45.734.361	45	/compat/linux/oracle
/dev/da1s1f	2.032.623	2	/compat/linux/sapmnt
/dev/da1s1g	2.032.623	2	/compat/linux/usr/sap

미리 Mylex 또는 PERC/3 RAID 소프트웨어로 두 개의 논리 드라이브를 설정하고 초기화한다. 이 소프트웨어는 BIOS 가 부팅하는 동안 시작된다.

SAP 는 오라클 서브 디렉터리에 나누어서 마운트 하도록 제안하지만 우리는 단순히 실제 서브 디렉터리에 생성하기로 결정했기 때문에 이 디스크 레이아웃은 SAP 권장 사항과 약간

다르다.

## 10.7.5.2 make world 와 새로운 커널

마지막 -STABLE 소스를 다운로드 한다. 커널 설정 파일을 설정한 후 world 와 사용자 커널로 다시 빌드한다. SAP R/3 와 오라클에 필요한 커널 매개 변수를 포함한다.

## 10.7.6 리눅스 환경 설치

### 10.7.6.1 리눅스 기본 시스템 설치

첫째 linux\_base 포트 설치가 필요하다(root 에서):

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

### 10.7.6.2 리눅스 개발 환경 설치

10.6 장에 따라 FreeBSD 에 오라클을 설치하려면 리눅스 개발환경이 필요하다:

```
# cd /usr/ports/devel/linux_devtools
# make install distclean
```

리눅스 개발 환경은 SAP R/3 46B IDES 설치를 위해서만 필요하다. FreeBSD 시스템에 오라클 DB 가 다시 링크되지 않는다면 필요없다. 이 환경은 리눅스 시스템에서 오라클 타볼(tarball)을 사용하는 경우다.

### 10.7.6.3 필요한 RPM 설치

R3SETUO 프로그램을 시작하려면 PAM(리눅스 인증 모듈) 지원이 필요하다. FreeBSD 4.3-STABLE 에 처음 SAP 를 설치하는 동안 우리는 필요한 모든 패키지와 PAM 설치를

시도했지만 결국 동작하는 PAM 패키지를 강제로 설치하였다. SAP R/3 4.6C SR2 에 우리는 동작하는 PAM RPM(레드햇 패키지 매니저)을 강제로 설치하였기 때문에 의존되는 패키지가 필요 없는 것으로 보인다:

```
# rpm -i --ignoreos --nodeps --root /compat/linux --dbpath /var/lib/rpm ₩
pam-0.68-7.i386.rpm
```

오라클 8.0.5 에 인텔리전트 에이전트를 실행하기 위해 우리는 RedHat Tcl 패키지 tcl-8.0.5-30.i386.rpm 도(설치하지 않으면 오라클을 설치하는 동안 링크가 동작하지 않는다) 설치했다. 오라클 링크에 관한 몇가지 다른 문제가 있지만 그것은 FreeBSD 가 아닌 리눅스와 오라클 문제다.

### 10.7.6.4 몇 가지 추가적인 힌트

/etc/fstab 에 linprocfs 를 추가하는 것도 좋은 아이디어다. 더 많은 정보는 linprocfs(5) 매뉴얼 페이지를 본다. 다른 매개변수 설정은 /etc/sysctl.conf 파일에 kern.fallback\_elf\_brand=3 을 설정한다.

## 10.7.7 SAP R/3 환경 생성

### 10.7.7.1 필요한 파일시스템과 마운트 포인트 생성

단순히 설치는 다음 파일시스템을 생성하면 된다:

마운트 포인트	GB 크기
/compat/linux/oracle	45 GB
/compat/linux/sapmnt	2 GB
/compat/linux/usr/sap	2 GB

몇 가지 링크생정도 필요하다. 링크를 생성하지 않으면 SAP 인스톨러는 생성된 링크를 체크하도록 요청한다:

```
# ln -s /compat/linux/oracle /oracle
```



그리고 다음 유저들도 필요하다:

유저 ID	유저 이름	일반 이름	그룹	추가적인 그룹	설명
1000	idsadm/prdadm	sidadm	sapsys	oper	SAP 관리자
1002	oraids/oraprd	Orasid	dba	oper	오라클 관리자

SAP 관리자"에 필요한 다음 엔트리를 adduser(8)로 추가한다:

```
Name: sidadm
Password: *****
Fullname: SAP Administrator SID
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/sidadm
Shell: bash (/compat/linux/bin/bash)
```

그리고 "오라클 관리자"에는 다음 엔트리가 필요하다:

```
Name: orasid
Password: *****
Fullname: Oracle Administrator SID
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/sid
Shell: bash (/compat/linux/bin/bash)
```

dba 와 oper 두 그룹을 사용한다면 그룹 oper 도 포함해야 된다.

### 10.7.7.3 디렉터리 생성



이들 디렉터리는 보통 분리된 파일시스템에 생성한다. 우리는 간단히 같은 RAID 5 에 디렉터리를 생성하기로 했다.

첫째 몇몇 디렉터리의 소유권과 권한을 설정한다(root 유저에서):

```
# chmod 775 /oracle
# chmod 777 /sapmnt
# chown root:dba /oracle
# chown sidadm:sapsys /compat/linux/usr/sap
# chmod 775 /compat/linux/usr/sap
```

두 번째로 유저 *orasid* 로 디렉터리를 생성한다. 이들 디렉터리는 모두 */oracle/SID* 의 서브디렉터리가 된다:

```
# su - orasid
# cd /oracle/SID
# mkdir mirrlogA mirrlogB origlogA origlogB
# mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 sapdata6
# mkdir saparch sapreorg
# exit
```

오라클 8.1.7 을 설치할 때 몇 개의 추가적인 디렉터리가 필요하다:

```
# su - orasid
# cd /oracle
# mkdir 805_32
# mkdir client stage
# mkdir client/80x_32
# mkdir stage/817_32
# cd /oracle/SID
# mkdir 817_32
```

**Note:** client/80x\_32 디렉터리는 정확하게 이 이름으로 사용된다. *x* 를 다른 번호로 바꾸지 않는다.

세 번째 단계에서 유저 *sidadm* 으로 디렉터리를 생성한다:



```
# su - sidadm
# cd /usr/sap
# mkdir SID
# mkdir trans
# exit
```

### 10.7.7.4 /etc/services 엔트리

SAP R/3 는 FreeBSD 에 설치할 때 정확하게 설정되지 않는 /etc/services 파일에 몇 가지 엔트리가 필요하다. 다음 엔트리를 추가한다(최소한 인스턴트 번호와 일치하는 엔트리가 필요하다. 이 경우 00을 추가하지만 dp, gw, sp와 ms에 00에서 99까지 모든 엔트리를 추가해도 문제는 없다). SAProuter 를 사용하거나 SAP OSS 에 접근해야 된다면 타겟 시스템에서 포트 3299 가 보통 SAProuter 에 사용되기 때문에 99도 필요하다:

sapdp00	3200/tcp # SAP Dispatcher.	3200 + Instance-Number
sapgw00	3300/tcp # SAP Gateway.	3300 + Instance-Number
sapsp00	3400/tcp #	3400 + Instance-Number
sapms00	3500/tcp #	3500 + Instance-Number
sapmsSID	3600/tcp # SAP Message Server.	3600 + Instance-Number
sapgw00s	4800/tcp # SAP Secure Gateway	4800 + Instance-Number

### 10.7.7.5 필요한 지역

SAP 는 RedHat 의 기본 설치에서 설정되지 않는 최소한 두 개의 지역이 필요하다. SAP 는 FTP 서버에서 필요한 RPM 다운로드를 제공한다(OSS 에 접근할 수 있는 소비자만 다운로드할 수 있는). 필요한 RPM 리스트는 노트 0171356 을 본다.

또한 적당한 링크를 생성하는 것도 가능하지만 업무용 시스템에는 권장하지 않는다(IDES 시스템과 문제없이 작동하지만). 다음 지역 설정이 필요하다:

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

따라서 다음과 같이 링크를 생성한다:

```
# cd /compat/linux/usr/share/locale
# ln -s de_DE de_DE.ISO-8859-1
# ln -s en_US en_US.ISO-8859-1
```

이들이 없다면 설치할 때 몇 가지 문제가 발생한다. 계속해서 무시한다면(파일 CENTRDB.R3S 에서 OK가 아닌 상태를 선택하여) 추가적인 노력 없이 SAP 시스템에 로그인하는 것은 불가능 하다.

### 10.7.7.6 커널 튜닝

SAP R/3 시스템은 수 많은 리소스가 필요하다. 그래서 커널 설정파일에 다음과 같은 매개변수를 추가했다:

```
# Set these for memory pigs (SAP and Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)"
# System V options needed.
options SYSVSHM #SYSV-style shared memory
options SHMMAXPGS=262144 #max amount of shared mem. pages
#options SHMMAXPGS=393216 #use this for the 46C inst.parameters
options SHMMNI=256 #max number of shared memory ident if.
options SHMSEG=100 #max shared mem.segs per process
options SYSVMSG #SYSV-style message queues
options MSGSEG=32767 #max num. of mes.segments in system
options MSGSSZ=32 #size of msg-seg. MUST be power of 2
options MSGMNB=65535 #max char. per message queue
options MSGTQL=2046 #max amount of msgs in system
options SYSVSEM #SYSV-style semaphores
options SEMMNU=256 #number of semaphore UNDO structures
options SEMMNS=1024 #number of semaphores in system
options SEMMNI=520 #number of semaphore identifiers
options SEMUME=100      #number of UNDO keys
```

SAP 의 문서에 최소값이 지정되어있다. 다만 리눅스를 위한 설명이 없기 때문에 더 많은 정보는 HP-UX 섹션(32-bit)을 본다. 4.6C SR2 를 설치하려는 시스템에 더 많은 메인 메모리가 있고 공유 세그먼트는 클수록 SAP 와 오라클에 좋지 때문에 거대한 양의 공유 메모리 페이지를 선택한다.

**Note:** *MAXDSIZ* 와 *DFLDSLZ* 을 최대 1GB 로 설정하고 i386 에 FreeBSD 4.5 를 기본적으로 설치한다. 그렇지 않으면 “ORA-27102: out of memory” 와 “Linux Error: 12: Cannot allocate memory” 같은 에러가 계속 발생한다.

## 10.7.8 SAP R/3 설치

### 10.7.8.1 SAP CDROM 준비

설치하는 동안 마운트와 언 마운트해야되는 여러 장의 CDROM 이 있다. 충분한 CDROM 드라이브가 있다면 모두 마운트한다. 우리는 CDROM 내용을 적당한 디렉터리에 복사했다:

```
/oracle/SID/sapreorg/cd-name
```

4.6B/IDES 를 설치할 때 *cd-name* 은 KERNEL, RDBMS, EXPORT1, EXPORT2, EXPORT3, EXPORT4, EXPORT5 와 EXPORT6 그리고 4.6C SR2 를 설치할 때 *cd-name* 은 KERNEL, RDBMS, DISK1, DISK2, DISK3, DISK4 와 LANG 중 하나다. CD 로 마운트된 모든 파일이름은 대문자를 사용해야 되고 그렇지 않으면 마운트에 *-g* 옵션을 사용한다. 다음 명령을 사용해 보자:

```
# mount_cd9660 -g /dev/cd0a /mnt  
# cp -R /mnt/* /oracle/SID/sapreorg/cd-name  
# umount /mnt
```

### 10.7.8.2 설치 스크립트 실행

첫째 install 디렉터리를 준비해야 된다:

```
# cd /oracle/SID/sapreorg
```

```
# mkdir install
# cd install
```

그리고 관련된 모든 파일을 install 디렉터리로 복사하는 설치스크립트를 시작한다:

```
# /oracle/SID/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

IDES 설치(4.6B) SAP R/3 데모 시스템에 완벽하게 튜닝 되어서 3 장의 EXPORT CD 대신 6 장이 있다. 여기서 IDES 중앙 인스턴스가 아닌 설치템플릿 CENTRDB.R3S 는 표준 중앙 인스턴스(R/3 와 데이터베이스) 위한 것이므로 EXPORT1 에서 정확한 CENTRDB.R3S 를 복사해야 되고 그렇지 않으면 R3SETUP 은 오직 3 개의 EXPORT CD 만 요청한다.

새로운 **SAP 4.6C SR2** 릴리즈는 4 장의 EXPORT CD 로 되어있다. 설치단계를 제어하는 매개변수 파일은 CENTRAL.R3S 다. 반대로 이전 릴리즈에는 데이터베이스가 있는 중앙 인스턴스 또는 데이터베이스가 없는 중앙 인스턴스를 위해 나누어진 설치템플릿이 없다. **SAP** 는 데이터베이스 설치에 분리된 템플릿을 사용한다. 그러나 나중에 설치를 다시 하겠다면 원본파일로 다시 시작하면 된다.

설치 중이나 설치 후에 **SAP** 는 전체 도메인네임이 아닌 컴퓨터 이름만 돌려받기 위해 hostname 이 필요하다. 그래서 호스트네임을 적절히 설정하거나 *orasid* 와 *sidadm* 에(또는 최소한 설치단계는 root 로 수행해야 되기 때문에 root 에) `alias hostname='hostname -s'` 로 엘리어스를 설정한다. **SAP** 를 설치하는 동안 두 유저에게 설치된 .profile 와 .login 파일을 수정하는 것도 가능하다.

### 10.7.8.3 R3SETUP 4.6B 시작

LD\_LIBRARY\_PATH 를 정확히 설정한다:

```
# export LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/exe:/oracle/805_32/lib
```

설치 디렉터리에서 root 로 R3SETUP 을 시작한다:

```
# cd /oracle/IDS/sapreorg/install
# ./R3SETUP -f CENTRDB.R3S
```

그러면 이 스크립트가 몇 가지 질문을 한다(기본값은 가로로 묶여있고 그 다음은 실제로 입력한 값이다):

질문	기본값	입력
Enter SAP System ID	[C11]	IDSEnter
Enter SAP Instance Number	[00]	Enter
Enter SAPMOUNT Directory	[/sapmnt]	Enter
Enter name of SAP central host	[troubadix.domain.d e]	Enter
Enter name of SAP db host	[troubadix]	Enter
Select character set	[1] (WE8DEC)	Enter
Enter Oracle server version (1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6		1Enter
Extract Oracle Client archive	[1] (Yes, extract)	Enter
Enter path to KERNEL CD	[/sapcd]	/oracle/IDS/sapreorg/KER NEL
Enter path to RDBMS CD	[/sapcd]	/oracle/IDS/sapreorg/RD BMS
Enter path to EXPORT1 CD	[/sapcd]	/oracle/IDS/sapreorg/EXP ORT1
Directory to copy EXPORT1 CD	[/oracle/IDS/sapreor g/CD4_DIR]	Enter
Enter path to EXPORT2 CD	[/sapcd]	/oracle/IDS/sapreorg/EXP ORT2
Directory to copy EXPORT2 CD	[/oracle/IDS/sapreor g/CD5_DIR]	Enter
Enter path to EXPORT3 CD	[/sapcd]	/oracle/IDS/sapreorg/EXP ORT3
Directory to copy EXPORT3 CD	[/oracle/IDS/sapreor g/CD6_DIR]	Enter
Enter path to EXPORT4 CD	[/sapcd]	/oracle/IDS/sapreorg/EXP ORT4
Directory to copy EXPORT4 CD	[/oracle/IDS/sapreor g/CD7_DIR]	Enter

Enter path to EXPORT5 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT5
Directory to copy EXPORT5 CD	[/oracle/IDS/sapreorg/CD8_DIR]	Enter
Enter path to EXPORT6 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT6
Directory to copy EXPORT6 CD	[/oracle/IDS/sapreorg/CD9_DIR]	Enter
Enter amount of RAM for SAP + DB		850Enter (in Megabytes)
Service Entry Message Server	[3600]	Enter
Enter Group-ID of sapsys	[101]	Enter
Enter Group-ID of oper	[102]	Enter
Enter Group-ID of dba	[100]	Enter
Enter User-ID of sidadm	[1000]	Enter
Enter User-ID of orasid	[1002]	Enter
Number of parallel procs	[2]	Enter

CD 를 다른 위치에 복사해 두지 않았다면 **SAP** 인스톨러는 필요한 CD 를 찾지 못해서(CD 에서 LABEL.ASC 파일로 확인되는) CD 를 넣고 마운트 후 확인하거나 마운트 경로를 입력하라고 요청한다.

CENTRDB.R3S 는 에러가 없을 것이다. 우리의 경우 EXPORT4 CD 를 다시 요구했지만 정확한 키를(6\_LOCATION 그리고 7\_LOCATION 등) 입력했기 때문에 정확한 값으로 계속 진행할 수 있었다.

아래에서 언급된 몇 가지 문제와 별개로 오라클 데이터베이스 소프트웨어가 설치되는 곳까지 진행한다.

#### 10.7.8.4 R3SETUP 4.6C SR2 시작

LD\_LIBRARY\_PATH 를 정확히 설정한다. 이것은 **오라클 8.0.5** 로 4.6B 를 설치하는 것과 다른 값이다:

```
# export LD_LIBRARY_PATH=/sapmnt/PRD/exe:/oracle/PRD/817_32/lib
```

설치 디렉터리에서 root 로 R3SETUP 시작한다:

```
# cd /oracle/PRD/sapreorg/install
# ./R3SETUP -f CENTRAL.R3S
```

이 스크립트는 몇 가지 질문을 한다(기본값은 가로로 묶여있고 그 다음은 실제로 입력한 값이다):

질문	기본 값	입력
Enter SAP System ID	[C11]	PRDEnter
Enter SAP Instance Number	[00]	Enter
Enter SAPMOUNT Directory	[/sapmnt]	Enter
Enter name of SAP central host	[majestix]	Enter
Enter Database System ID	[PRD]	PRDEnter
Enter name of SAP db host	[majestix]	Enter
Select character set	[1] (WE8DEC)	Enter
Enter Oracle server version (2) Oracle 8.1.7		2Enter
Extract Oracle Client archive	[1] (Yes, extract)	Enter
Enter path to KERNEL CD	[/sapcd]	/oracle/PRD/sapreorg/KERNEL
Enter amount of RAM for SAP + DB	2044	1800Enter (in Megabytes)
Service Entry Message Server	[3600]	Enter
Enter Group-ID of sapsys	[100]	Enter
Enter Group-ID of oper	[101]	Enter
Enter Group-ID of dba	[102]	Enter
Enter User-ID of oraprd	[1002]	Enter
Enter User-ID of prdadm	[1000]	Enter
LDAP support		3Enter (no support)
Installation step completed	[1] (continue)	Enter
Choose installation service	[1] (DB inst,file)	Enter

설치하는 동안 OSUSERDBSID\_IND\_ORA(유저 orasid 생성) 그리고 OSUSERSIDADM\_IND\_ORA(유저 sidadm 생성) 구문에서 에러가난다면 유저를 생성한다.



아래에서 언급된 몇 가지 문제와 별개로 오라클 데이터베이스 소프트웨어가 설치되는 곳까지 진행한다.

## 10.7.9 오라클 8.0.5 설치

리눅스와 오라클에서 가능한 문제에 대해서는 적당한 SAP 노트와 오라클 readme 를 본다. 보통 호환되지 않는 라이브러리 문제는 없다.

오라클 설치에 대한 더 많은 정보는 오라클 설치 첵터를 참고한다.

### 10.7.9.1 오라클 8.0.5 와 orainst 설치

오라클 8.0.5 가 사용된다면 오라클 8.0.5 가 예전의 glibc(RedHat 6.0)와 링크된 것처럼 성공적으로 링크를 생성하기 위해 몇 가지 추가적인 라이브러리가 필요하지만 RedHat 6.1 은 새로운 glibc 를 사용하고 있다. 그래서 링크가 동작하도록 다음의 추가적인 패키지를 설치해야 된다:

```
compat-libs-5.2-2.i386.rpm  
  
compat-glibc-5.2-2.0.7.2.i386.rpm  
  
compat-egcs-5.2-1.0.3a.1.i386.rpm  
  
compat-egcs-c++-5.2-1.0.3a.1.i386.rpm  
  
compat-binutils-5.2-2.9.1.0.23.1.i386.rpm
```

더 많은 정보는 적당한 SAP 노트나 오라클 Readme 를 본다. 옵션이 없다면(설치하는 동안 우리는 체크할 수 있는 충분한 시간이 없었다) 원본 바이너리를 사용할 수 있거나 원래 RedHat 시스템에서 링크된 바이너리를 사용한다.

인텔리전트 에이전트를 컴파일 하려면 RedHat Tcl 패키지가 설치되어 있어야 한다. tcl-8.0.3-20.i386.rpm 이 없다면 RedHat 6.1 의 tcl-8.0.5-30.i386.rpm 같은 새로운 것도 사용할 수 있다.

다시 링크하는 부분에서 설치는 상당히 직관적이다:

```
# su - oraid  
# export TERM=xterm  
# export ORACLE_TERM=xterm  
# export ORACLE_HOME=/oracle/IDS  
# cd /ORACLE_HOME/orainst_sap  
# ./orainst
```

리눅스에서 현재 이용할 수 없기 때문에 *Oracle On-Line Text Viewer*를 제외하고 소프트웨어가 설치될 때까지 Enter를 눌러 모두 확인한다. 그리고 오라클은 유효한 gcc, egcs 나 i386-redhat-linux-gcc 대신 gcc, egcs 또는 i386-redhat-linux-gcc와 링크되어야 한다.

시간 때문에 우리는 **오라클 8.0.5 PreProduction** 릴리즈 바이너리를 사용하기로 했다. RDBMS CD에서 버전을 받으려는 첫 번째 이후의 시도는 실패했고 그 당시 정확한 RPM을 찾는 것은 우리가 많았다.

## 10.7.9.2 리눅스용(커널 2.0.33) 오라클 8.0.5 Pre-production

### 릴리즈 설치

이 설치의 상당히 쉽다. CD를 마운트하고 인스톨러를 시작한다. 오라클 홈 디렉터리의 위치를 물어보고 모든 바이너리를 홈 디렉터리에 복사한다. 이전에 설치한 RDBMS가 남아있었지만 나중에 오라클 데이터베이스는 아무런 문제없이 시작할 수 있었다.

## 10.7.10 오라클 8.1.7 리눅스 타볼 설치

oracle81732.tgz 타볼을 가지고 리눅스 시스템의 설치 디렉터리 /oracle/S/D/817\_32/에 타볼을 푼다.

### 10.7.11.1 SAP R/3 설치 계속

첫째로 유저 `idsamd(sidadm)`과 `oraids(orasid)`의 설정 환경을 체크한다. 이들은 모두 `hostname` 을 사용하는 `.profile`, `.login` 과 `.cshrc` 를 가지고 있다. 이 경우 파일 3 개의 모든 `hostname` 을 완벽하게 유효한 이름 `hostname -s` 로 변경해야 된다.

### 10.7.11.1 데이터베이스 로드

`exit` 를 선택하거나 안 하느냐에 따라 `R3SETUP` 를 재 시작 하거나 계속 진행할 수 있다. `R3SETUP` 는 테이블 스페이스를 생성하고 데이터와(46B IDES 는 EXPORT1 에서 EXPORT6 까지 그리고 46C 는 DISK1 에서 DISK4 까지) `R3load` 를 데이터베이스에 넣어서 로드한다.

데이터베이스 로드가 끝났을 때(몇 시간 정도가 필요할 것이다) 어떤 패스워드가 요구된다. 테스트 설치를 위해 유명한 기본 패스워드를 사용할 수 있다(보안 문제가 있다면 다른 것을 사용한다):

질문	입력
Enter Password for sapr3	sapEnter
Confirum Password for sapr3	sapEnter
Enter Password for sys	change_on_installEnter
Confirm Password for sys	change_on_installEnter
Enter Password for system	managerEnter
Confirm Password for system	managerEnter

여기서 우리가 4.6B 를 설치하는 동안 `dipgntab` 에서 몇 가지 문제가 있었다.

### 10.7.11.2 리스너

다음과 같이 유저 `orasid`로 오라클 리스너를 시작한다:

```
% umask 0; lsnrctl start
```

그렇지 않으면 소켓에 정확한 퍼미션이 할당되지 않기 때문에 `ORA-12546` 에러를 보게 된다. 자세한 사항은 SAP 노트 072984 를 본다.

### 10.7.11.3 MNLS 테이블 업데이트

Latin-1 이 아닌 언어를 SAP 시스템에 탑재하기를 원치 않는다면 여러 나라의 언어지원 테이블을 업데이트 한다. 이 사항은 SAP OSS 노트 15023 과 45619 에 설명되어 있다. 그렇지 않으면 SAP 를 설치하는 동안 이 질문을 건너뛰어도 된다.

**Note:** MNLS 가 필요 없다면 테이블 TCPDB 를 체크해야 되고 이것이 끝나지 않는다면 초기화 해야 된다. 더 많은 정보는 SAP 노트 0015023 과 0045619 를 본다.

## 10.7.12 설치 후 단계

### 10.7.12.1 SAP R/3 라이선스 키 요청

SAP R/3 라이선스 키를 요청해야 된다. 설치 중에 지정한 라이선스는 4 주 동안만 일시적으로 유효하므로 키를 요청해야 된다. 첫째로 하드웨어 키를 받는다. 유저 idsadm 으로 로그인해서 saplicense 를 실행한다:

```
# /sapmnt/IDS/exe/saplicense -get
```

saplicense 를 매개변수 없이 실행하며 옵션 리스트를 보여준다. 받은 라이선스 키에 따라 설치할 수 있다:

```
# /sapmnt/IDS/exe/saplicense -install
```

다음 값을 입력해야 된다:

SAP SYSTEM ID	= <i>SID, 3 chars</i>
CUSTOMER KEY	= <i>hardware key, 11 chars</i>
INSTALLATION NO	= <i>installation, 10 digits</i>
EXPIRATION DATE	= <i>yyyymmdd, usually "99991231"</i>
LICENSE KEY	= <i>license key, 24 chars</i>

### 10.7.12.2 유저 생성

클라이언트 000 에서 유저를 생성한다(이 작업을 끝내기 위해 클라이언트 000 에서 몇 가지 작업이 필요하지만 유저 sap\*와 ddic 는 다르다). 유저 네임에 우리는 보통 wartung(또는 영어에서 service)를 선택했다. 필요한 프로파일은 *sap\_new*와 *sap\_all*이다. 기본 유저 패스워드의 추가적인 안전을 위해 모든 클라이언트를 변경해야 된다(이것은 유저 sap\*와 ddic 를 포함한다).

### 10.7.12.3 전송 시스템, 프로파일, 운용 모드 등 설정

ddic 및 sap\*와 다른 유저는 클라이언트 000 에서 최소한 다음과 같은 작업이 필요하다:

작업	트랜잭션
전송 시스템 설정, 예: Stand-Alone Transport Domain Entity	STMS
시스템을 위한 프로파일 생성/수정	RZ10
운영 모드와 인스턴스 유지	RZ04

이들과 다른 모든 설치 후 단계는 **SAP** 설치 가이드에 완벽하게 설명되어있다.

### 10.7.12.4 initsid.sap(initIDS.sap) 편집

파일 /oracle/IDS/dbs/initIDS.sap 는 **SAP** 백업 프로필을 가지고 있다. 여기서 사용되는 테잎 크기와 압축된 테잎 등은 정의가 필요하다. sapdba/brbackup 으로 실행되는 동안 정의하기 위해 우리는 다음 값을 변경하였다:

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

설명:

*compress*: 우리가 사용하는 테잎은 하드웨어 압축을 하는 HP DLT1 이다.

*archive\_function*: 이 옵션은 오라클 아카이브 로그를 저장하기 위한 기본동작을 정의한다: 새로운 로그파일은 테잎에 저장하고 이미 저장된 로그는 다시 한번 저장된 후 삭제된다. 데이터베이스를 복구해야될때 이 옵션은 아카이브 테잎 중 하나가 이상하더라도 수많은 문제를 방지한다.

*cpio\_flags*: 기본값은 블록크기를 5120 Bytes 로 설정하는 *-B*를 사용한다. HP 는 DLT 테잎에 최소한 32K 블록크기를 권장하기 때문에 우리는 64K 로 설정하도록 *--block-size=128*을 사용한다. inode 번호가 65535 보다 크기 때문에 *--format=newc*가 필요하다. 마지막 옵션 *-quiet*가 필요하고 사용하지 않는다면 *brbackup* 이 *cpio* 저장된 블록번호를 출력하자마자 요청할 것이다.

*cpio\_in\_flags*: 플래그는 테잎에서 데이터를 다시 로딩하기 위해 필요하다. 포맷은 자동으로 감지된다.

*tape\_size*: 이 옵션은 보통 저 용량 테잎에 사용된다. 보안적인 이유로(우리는 하드웨어 압축을 사용한다) 이 값은 실제 값보다 약간 낮다.

*tape\_address*: rewind 할 수 없는 장치는 *cpio* 로 사용한다.

*tape\_address\_rew*: rewind 할 수 있는 장치는 *cpio* 로 사용한다.

### 10.7.12.5 설치 후 설정 문제

다음 **SAP** 매개변수를 설치 후에 켜야 된다(예를 들어 IDES 46B 는 1GB 메모리):

이름	값
ztta/roll_extension	250000000
abap/heap_area_dia	300000000
abap/heap_area_nondia	400000000
em/initial_size_MB	256
em/blocksize_kB	1024

ipc/shm_psize_40	70000000
------------------	----------

SAP Note 0013026:

이름	값
ztta/dynpro_area	2500000

SAP Note 0157246:

이름	값
rdisp/ROLL_MAXFS	16000
rdisp/PG_MAXFS	30000

**Note:** 위의 매개변수로 1GB의 메모리를 가지고 있는 시스템은 다음과 비슷한 메모리 소비량을 볼 수 있다:

Mem: 547M Active, 305M Inact, 109M Wired, 40M Cache, 112M Buf, 3492K Free

## 10.7.13 설치 중 문제

### 10.7.13.1 문제 해결 후 R3SETUP 재 시작

에러가 발생했다면 R3SETUP을 정지한다. 정확한 로그파일을 보고 에러를 해결했다면 R3SETUP을 다시 시작한다. 보통 마지막 단계에서 REPEAT를 선택한 것에 대해 R3SETUP가 문제를 제기한다.

R3SETUP를 재 시작하려면 적당한 R3S 파일로 시작한다:

```
# ./R3SETUP -f CENTRDB.R3S
```

4.6B는

```
# ./R3SETUP -f CENTRAL.R3S
```

4.6C 는 CENTRAL.R3S 나 DATABASE.R3S 에서 에러가 발생하더라도 문제가 없다.

**Note:** 어떤 단계에서 R3SETUP 는 데이터베이스와 **SAP** 프로세스가 운용 중 이라고 간주한다(이러한 단계가 이미 끝났기 때문에). 예를 들어 데이터베이스가 시작되지 않았다면 에러가 발생한다. 따라서 에러를 수정한 후 R3SETUP 를 다시 시작하기 전에 데이터베이스와 **SAP** 를 수동으로 시작해야 된다.

오라클 리스너도 정지되었다면(예를 들면 시스템 재 부팅이 필요했다면) 오라클 리스너도 다시 시작하는 것을 잊지 않는다(*orasid*에서 `umask 0; lsnrctl start`).

### 10.7.13.2 R3SETUP 동안 OSUSERSIDADM\_IND\_ORA

이 단계에서 R3SETUP 이 문제가 발생하면 R3SETUP 이 이때 사용하는 템플릿 파일을 수정한다(CENTRDB.R3S(4.6B), CENTRAL.R3S 또는 DATABASES.R3S(4.6C)).

[*OSUSERSIDADM\_IND\_ORA*] 위치의 내용이나 *STATUS=ERROR* 엔트리만 찾아서 다음 값으로 수정한다:

```
HOME=/home/sidadm (was empty)
STATUS=OK (had status ERROR)
```

그리고 R3SETUP 를 다시 시작한다.

### 10.7.13.3 R3SETUP 동안 OSUSERDBSID\_IND\_ORA

R3SETUP 은 이 단계에서도 문제가 발생할 수 있다. 여기 에러는 구문 OSUSERSIDADM\_IND\_ORA 과 비슷하다. 이때 R3SETUP 이 사용하는 템플릿 파일을 수정한다(CENTRDB.R3S(4.6B)또는 CENTRAL.R3S 또는 DATABASES.R3S(4.6C)).

[*OSUSERDBSID\_IND\_ORA*] 위치의 내용이나 *STATUS=ERROR* 엔트리를 찾아서 다음 값으로 수정한다:

```
STATUS=OK
```



그리고 R3SETUP 를 다시 시작한다.

#### 10.7.13.4 오라클을 설치하는 동안 “oraview.vrf FILE NOT FOUND”

설치를 시작하기 전에 *Oracle On-Line Text Viewer* 선택을 해제하지 않았다. 리눅스용에서 현재 사용할 수 없더라도 이 옵션은 설치에서 표시되어있다. 오라클 설치메뉴에서 이 제품 선택을 해지하고 설치를 계속한다.

#### 10.7.13.5 R3SETUP, RFC 또는 SAPgui 를 시작하는 동안 “TEXTENV\_INVALID”

이 에러가 발생했다면 정확한 지역이 빠진 것이다. SAP 노트 0171356 에는 설치가 필요한 RPM 을 나열하고 있다(예: RedHat 6.1 의 saplocales-1.0-3, saposcheck-1.0-1.) 이 경우 관련된 모든 에러를 무시하고 R3SETUP 이 문제를 표시할 때마다 **ERROR** 을 **OK** 로 적당히 설정하고 R3SETUP 를 다시 시작한다. **SAP** 시스템은 정확하게 설정되지 않았기 때문에 시스템이 시작되었어도 SAPgui 로 시스템에 연결할 수 없다. 주어진 다음 메시지를 따르고 예전 리눅스 **SAPgui** 로 연결을 다시 시도한다:

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trmsgso. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occured [trmsgsi. 0410]
*** ERROR => Error when generating text environment. [trmsgsi. 0435]
*** ERROR => function failed [trmsgsi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```

이 동작은 **SAP R/3** 에 지역할당이 정확히 되지 않았고 적절한 설정도 필요하다(몇몇 데이터베이스 테이블에서 빠진 엔트리). **SAP** 와 연결하도록 DEFAULT.PFL 파일에(노트 0043288 을 본다) 다음 엔트리를 추가한다:

```
abap/set_etct_env_at_new_mode = 0
install/collate/active = 0
rscp/TCP0B = TCP0B
```

SAP 시스템을 다시 시작한다. 지역 지정과 언어 설정이 동작하지 않더라도 이제 시스템에 연결할 수 있다. 국가설정을(그리고 정확한 지역을 지원하는) 정확히 한후 이러한 엔트리를 DEFAULT.PFL 에서 삭제할 수 있고 SAP 시스템을 재 시작할 수 있다.

### 10.7.13.6 ORA-00001

위 에러는 FreeBSD 4.5 의 오라클 8.1.7 에서만 발생하는 에러다. 오라클 데이터베이스를 적절히 초기화하지 못하고 충돌했기 때문에 시스템의 세마포어와 공유 메모리를 제거한다. 그 다음 데이터베이스를 시작하고 ORA-00001 을 입력한다.

ipcs -a 로 이들을 찾고 ipcrm 으로 삭제한다.

### 10.7.13.7 ORA-00445 (백그라운드 프로세스 PMON 이 시작되지 않았다)

이 에러는 오라클 8.1.7 에서 발생한다. 이 에러는 데이터베이스가 유저 prdadm 에서 startsap 스크립트로(예를 들면 startsap\_majestix\_00) 시작되었을 때 발생한다.

가능한 방법은 svrmgrl 대신 유저 oraprd 로 데이터베이스를 시작한다:

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> startup;
SVRMGR> exit
```

### 10.7.13.8 ORA-12546 (정확한 퍼미션으로 리스너 시작)

유저 oraids 에서 다음 명령으로 오라클 리스너를 시작한다:

```
# umask 0; lsnrctl start
```

그렇지 않으면 소켓에 정확한 퍼미션이 할당되지 않기 때문에 ORA-12546 에러를 보게 된다. SAP 노트 0072984 를 본다.

### 10.7.13.9 ORA-27102 (메모리 부족)

이 에러는 *MAXDSIZ*와 *DFLDSIZ* 값이 1GB 보다 큰 값을 사용하려고 할 때(1024x1024x1024) 발생한다. 추가적으로 우리는 "Linux Error 12: Cannot allocate memory"를 보았다.

### 10.7.13.10 R3SETUP 동안 [DIPGNTAB\_IND\_IND]

일반적으로 SAP 노트 0130581 을 본다 (R3SETUP 단계에서 *DIPGNTAB*이 끝난다). IDES-기능을 설치하는 동안 몇 가지 이유로 설치 단계가 적절한 **SAP** 시스템네임 "IDS"를 사용하지 못하고 빈 문자열 ""을 사용한다. 경로는 *SID*를(여기서는 IDS) 동적으로 사용하여 생성되기 때문에 디렉터리 접근과 관련된 몇 가지 사소한 문제가 발생한다. 그래서 다음 경로 대신:

```
/usr/sap/IDS/SYS/...  
/usr/sap/IDS/DVMGS00
```

아래의 경로를 사용한다:

```
/usr/sap//SYS/...  
/usr/sap/D00
```

설치를 계속하기 위해 우리는 추가적인 디렉터리에 링크를 생성했다:

```
# pwd  
/compat/linux/usr/sap  
# ls -l  
total 4
```



방해한다. 이 결과로 문제가 생겨도 스크립트 자체가 에러를 보여주지 못한다.

SAP 프로세스가 정확히 시작되었는지 체크하기 위해 모든 **오라클**과 **SAP** 프로세스 상태를 보여주는 `ps ax | grep SID`로 살펴본다. **SAP** 시스템에 연결할 수 없고 어떤 프로세스를 잃어버렸다면 `/usr/sap/SID/DEVBMGSnr/work/`에서 찾을 수 있는 적절한 로그파일을 살펴본다. 이 파일은 `dev_ms`와 `dev_disp`에서 찾을 수 있다.

**오라클**과 **SAP**가 사용하는 공유메모리의 양이 커널 설정파일에 정의한 양을 초과했다면 31 신호가 발생하기 때문에 더 큰 값으로 해결할 수 있다:

```
# larger value for 46C production systems:
options SHMMAXPGS=393216
# smaller value sufficient for 46B:
#options SHMMAXPGS=262144
```

### 10.7.13.14 saposcol 시작 실패

프로그램 `saposcol`과 몇 개의 문제가(버전 4.6D) 있다. **SAP** 시스템은 시스템성능에 관련된 데이터를 수정하기 위해 `saposcol`을 사용한다. 이 프로그램을 **SAP** 시스템에 사용할 필요가 없기 때문에 이 문제는 사소한 문제가 될 수 있다. 이전 버전(4.6B)은 동작하지만 모든 데이터를(예를 들면 CPU 사용량과 같은 많은 콜은 단지 0을 되돌려준다) 수집하지는 않는다.

## 10.8 발전된 주제

어떻게 리눅스 바이너리 호환성이 작동하는지 궁금하다면 이 섹션에 원하는 설명이 있다.

### 10.8.1 어떻게 동작하는가?

FreeBSD는 “execution class loader”라는 추상적 개념을 가지고 있다. 이것이 `execve(2)` 시스템 콜의 발단이 됐다.

셸 인터프리터나 셸 스크립트를 실행하기 위해 FreeBSD는 `#!` 로더에 의해 저장된 싱글 로더 대신에 로더 리스트를 가지고 있다. 역사적으로 유닉스 플랫폼의 유일한 로더는

시스템이 알고 있는 바이너리인지 확인해서 바이너리라면 바이너리 로더를 실행하는 매직번호(magic number)를(보통 파일의 처음 4 또는 8 바이트) 확인한다. 시스템의 바이너리 종류가 아니라면 `execve(2)`가 실패를 되돌려주고 셸이 셸 스크립트처럼 실행한다.

이 가정은 현재의 셸이 무엇이든 간에 기본이다.

나중에 첫 번째 2 캐릭터를 확인하도록 `sh(1)`에 `hack` 이 만들어졌고 2 캐릭터가 `:#n` 이면 `csh(1)` 셸을 대신 실행한다(SCO 가 처음으로 이 `hack` 을 만들었다고 생각된다).

이제 FreeBSD 가 하는 것은 로더 리스트를 거치는 것이다. 이것은 인터프리터에 대해 알고 있는 일반적인 `#!` 로더인데 이들은 공백들에 의해 구분되는 문자들과 `/bin/sh` 에 저장된 내용들로 구성된다.

리눅스 ABI 를 지원하기 위해 FreeBSD 는 ELF 바이너리로(FreeBSD, Solaris™, 리눅스 등 ELF 이미지 타입을 가지고 있는 다른 OS 는 차이가 없다) 매직번호를 확인한다.

SVR4/Solaris ELF 바이너리에 없는 ELF 로더는 ELF 이미지 섹션에 설명되어있는 특화된 *brand* 를 찾는다.

리눅스 바이너리가 동작하도록 이들을 `brandelf(1)`에서 Linux 형식으로 `brand` 해야 된다:

```
# brandelf -t Linux file
```

`brand` 가 끝나면 ELF 로더는 파일에서 *Linux brand* 를 찾을 수 있다.

ELF 로더가 *Linux brand* 를 확인하면 로더는 *proc* 구조의 포인터를 변경한다. 모든 시스템 콜은 이 포인터로 인덱스 된다(전통적인 유닉스 시스템에서 이것은 시스템 콜을 포함하고 있는 `sysent[]` 구조배열이다). 게다가 트램폴린(trampoline) 코드 신호의 trap vector 를 특별히 제어하는 플래그가 프로세스에 할당되고 리눅스 커널 모듈이 제어하는 몇 가지를 수정했다.

리눅스 시스템 콜 벡터는 주소가 커널 모듈에 있는 `sysent[]` 엔트리 리스트를 가지고 있다.

리눅스 바이너리가 시스템 콜을 호출할 때 내장된 코드는 시스템 콜 기능 포인터가 *proc* 구조를 참조하지 않게 하고 FreeBSD 가 아닌 리눅스 시스템 콜 엔트리 포인트를 갖게 한다.

게다가 리눅스 모드는 동적으로 *reroots* 를 검색한다; 이 결과로 파일시스템을 마운트할 때 *union* 옵션이 파일시스템 마운트에 사용된다(*unionfs* 파일시스템 타입이 아니다). 처음 */compat/linux/original-path* 디렉터리에서 파일을 검색하고 검색에 실패하면 */original-path* 디렉터리에서 검색은 끝난다. 이것은 다른 바이너리에 필요한 바이너리를 실행할 수 있다(예: 리눅스 툴 체인은 리눅스 ABI 지원에서 모두 실행할 수 있다). 적당한 리눅스 바이너리가 없다면 리눅스 바이너리는 FreeBSD 바이너리를 로드해서 실행할 수 있고, 리눅스에서 리눅스 바이너리가 실행되지 않았다고 경고를 보내지 못하도록 *uname(1)* 명령을 */compat/linux* 디렉터리에 둘 수 있다.

이 결과로 FreeBSD 커널에 리눅스 커널이 있다; 커널과 FreeBSD 시스템 콜 테이블 엔트리 그리고 리눅스 시스템 콜 테이블 엔트리가 제공하는 모든 서비스를 실행하는 파일시스템 운용, 가상 메모리 운용, 신호 전달, System V IPC 등 다양하고 근본적인 기능이 있다. 유일한 차이점은 FreeBSD 바이너리는 FreeBSD *glue* 기능을 가지고 있고 리눅스 바이너리는 리눅스 *glue* 기능을 가지고 있다(대부분의 예전 OS 는 자신만의 *glue* 기능을 가지고 있다.

**Glue:** 프로세스가 만드는 콜의 *proc* 구조에서 동적으로 초기화된 포인터를 끄는, 참조되지 않는 주소기능 대신 *static global sysent[]* 구조 배열의 주소기능

어느 것이 진짜 FreeBSD ABI 인지 문제되지 않는다. 기본적으로 유일한 차이점은(현재: 이것은 새로운 릴리즈에서 쉽게 변경할 수 있고 아마도 다음 버전부터 가능할 것이다) FreeBSD *glue* 기능은 정적으로 커널에 링크되어있고 리눅스 *glue* 기능도 정적으로 링크 시키거나 커널 모듈을 통해 접근할 수 있다.

그러나 이 기능은 사실 에뮬레이션이 아닌가? 이것은 에뮬레이션이 아니고 ABI 기능이다. 에뮬레이터는(또는 시뮬레이션) 관련이 없다.

그렇다면 왜 가끔 “리눅스 에뮬레이션” 이라고 하는가? 이 용어가 FreeBSD 의 이미지를 버렸다. 그 당시 이 기능을 표현할 수 있는 적절한 단어가 없었다; 코드를 컴파일 하지 않거나 모듈을 로드하지 않는다면 FreeBSD 가 리눅스 바이너리를 실행하는 것은 거짓이어서 로드되는 것을 설명하기 위한 단어가 필요했고 따라서 “리눅스 에뮬레이터”가 탄생했다.