Informix-SE 7.24.UC5을 기준으로 한 리눅스용 인포믹스에 대한 자주 나오는 질문과 답변

Kurt Wall, kwall@uswest.net v0.4, 1998년 11월 2일

장동수, jdongsu@untel.co.kr 1999년 4월 21일

이 문서는 리눅스용 **Informix Standard Engine**과 **ESQL/C** 개발 도구의 설치, 설정 및 사용에 관한 자주 나오는 질문(FAQs)과 답변 모음이다. 받드시, 작업이 진행 중임을 고려해야만 한다. 이 문서는 리눅스에 인포믹스-SE를 설치하고 실행할 때 가장 자주 접하게되는 장애물에 대해서 언급한다. ODBC나 ESQL/C같은 특정한 부분에 대해서는 취약하다. 여기에는 관계형 데이터베이스 이론에 대한 입문서가 아니다. 이 주제만으로도 모든 경력과 수십권의 책을 바쳐야 한다. 형편이 허락하면, 나는 이러한 책들과 관련된 지침들을 추가하고 싶다.

목 차

제 1 절 개요

1.1 개정 역사

1.1.1 버젼 0.4

1998년 11월 2일

1. 앞으로 나오게 될 인포믹스 제품군에 대한 새소식 추가
2. glibc 버전에 대한 정보 추가
3. "sqlexec와 sqlexecd의 차이점은 무엇인가?"에 대한 답변

4. informix 사용자와 그룹 계정 만들기에 대한 섹션 재작성
5. future-proofing 추가
6. 버그 번호 추가
7. "약어" 섹션 추가
8. 멀티프로세서와 쓰레드에 관련된 진보된 논의 추가
9. 추가적인 리소스 수정 및 추가
10. NFS마운트된 데이터베이스가 나쁜 이유에 대한 집중분석
11. 사소한 구조적 변경
12. 타이핑 오류 수정

1.1.2 버젼 0.3

최초의 일반 배포 - 1998년 9월 25일

1. sqlexec 좀비에 대한 명백한 문제점들
2. PHP에 대한 지침 추가
3. NFS에 대한 새로운 질문
4. 사소한 구조적 변경
5. 타이핑 오류 수정

1.1.3 버젼 0.2

최초의 배포판 - 1998년 9월 20일

1.1.4 버젼 0.1

초고 버젼 - 1998년 9월 18일 `comp.databases.informix` 과 linux-informix IIUG 메일링 리스트에만 게시함.

1.2 Disclaimer

1.3 배포

이 FAQ는 필요하다고 판달될 때, 그리고 내가 시간이 날 때(거의 ^^;) 개정될것이다. 나는 대략 매달 그렇게 하려고 한다. 이것은 아래의 주소에서 온라인으로 볼 수 있을 것이다:

- `http://www.iug.org/techinfo/linux/linux\_faq.html`
- `http://www.users.uswest.net/\~{}kwall/iolfaq.html`
- `http://www.intrware.com/ms/itwr/pfaq/ifmx/linuxfaq/iolfaq.html`

이 FAQ를 다른 언어로 번역하려는 제안이나 노력을 반가운 일이다. 현재 다음 언어들에 대한 번역이 있거나 진행중이다:

- 일본어 - 작업 중. 더 자세한 정보는 Norihiro Gouto (`ngotou@geocities.co.jp`) 에게 연락할것.
- 스페인어 - 작업 중. 더 자세한 정보는 Alejandro Fernandez (`afernand@lanet.com.ar`) 에게 연락할 것.
- 루마니아어 - 작업 중. 더 자세한 정보는 Sebastian Paul Avarvarei (`proteus@tearwarent.internal.romus.com`) 에게 연락할 것.
- 독일어 - 할 사람을 기다리는 중

이것을 얻을 수 있는 익명 FTP의 주소는:

- `http://www.users.uswest.net/\~{}kwall/iolfaq.tar.gz`

또한, 나는 일반적인 리눅스 웹과 FTP사이트에 게시되길 바란다. Intraware 도 이 FAQ (Dan에게 감사한다)를 미러링하고, 그들이 보내는 등록 전자우편에 이것에 대한 링크를 포함할 예정이다.

1.3.1 사용된 약어

- **IANAL** - I Am Not A Lawyer
- **TANSTAAFL** - There Ain't No Such Thing As A Free Lunch
- **IDN** - Informix Developers Network
- **IDS** - Informix Dynamic Server
- **IIUG** - International Informix Users Group
- **RDBMS** - Relational DataBase Management System
- **YMMV** - Your Mileage May Vary

제 2 절 소프트웨어 구하기

2.1 지원되는 플랫폼과 지원되지 않는 플랫폼

이 FAQ를 가장 최근에 배포할때까지, Informix-SE가 정식으로 테스트된 플랫폼(모두 인텔)은 다음과 같다: 테스트 되었다:

- Caldera OpenLinux 1.2 (kernel 2.0.33, libc5)

- S.u.S.E. 5.2 (libc5)
- RedHat 5.1 (kernel 2.0.34, glibc 2.0.7)

glibc버전의 릴리즈 노트에는 Pentium II 400 Mhz상에서 작동하는 RedHat 5.1에서 테스트되었으며, libc5 버전의 경우에는 Pentium II 400 Mhz에서 작동하는 OpenLinux 1.2에서 테스트되었다고 한다.

인포믹스에는 현재 두 개의 분리되는 포트가 있다. 원래 버전은 libc5로 컴파일되었고 Caldera의 OpenLinux 1.2, S.u.S.E. 5.2, Slackware 3.5와 이전 버전, Debian 1.3 같은 배포판에서 변경없이 실행되었다. 인포믹스는 새로운 glibc(libc6이라 불렀던)을 지원을 약속했고 RedHat 사용자도 그것을 사용할 수 있게될 것이다. 인포믹스는 10월 마지막 주 중에 그 약속을 이행할 것이다.

glibc/libc6 기반 시스템을 사용하는 시스템들도 대부분 libc5 런타임을 설치하고 libc5 버전을 실행할 수 있을 것이다. 이것은 지원되지 않는 설정이므로 당신의 시스템이 망가지더라도 나를 원망하지 마라. 그러한 시스템에서 무사히 실행하고 있다는 몇몇의 보고가 있지만 *위험부담은 당신이 지는 것이다(caveatemptor)*. glibc/libc6 시스템에 libc5를 설치하는 방법은 이 문서의 끝에 있는 10(추가적인 리소스) 를 참조하라.

[역자주: 이 문서를 번역하는 시점에서 SE 7.24UC5가 발표되었으며, glibc기반의 알짜 RH5.2에서 잘 작동하고 있음을 확인했다.]

2.2 Intraware에 등록하기

주: 만일 당신이 사는 곳이 미국이 아니라면, 당신은 800-331-1763에 전화해야 한다. 미국 밖에서 800 번을 다이얼하는 것이 항상 가능한 것은 아니라고 들었는데, 어떻게하든 가장 가까운 인포믹스 사무실에 연락할 수 있을 것이다. 위치 목록은 `http://www.informix.com/informix/contact/offices`을 방문할 것.

Informix-SE를 다운로드 받을려면, 당신은 먼저 인포믹스의 다운로드를 취급하는 Intraware 사이트에 등록해야 한다. 등록은 무료이고, 질문도 그다지 까다롭지 않다. 등록하기 위해서는 `http://www.intraware.com/idn/linux/index.html`에 가서 지시를 따르면 된다. 당신은 Intraware로 부터 몇 시간 내에 사용자코드와 암호가 담긴 메일을 받게 될 것이다(나는 대략 20분만에 받았다). 이 정보를 인쇄해두고 백업본을 만들어 두어라.

2.3 소프트웨어 구하기

2.3.1 CD 주문하기

CD를 주문하려면 800-331-1763에 전화할 것.

2.3.2 Intraware에서 다운로드 받기

Intraware 사용자코드와 암호를 받은 다음, `http://www.intraware.com/idn/linux/index.html`에서 "Download Page" 링크를 클릭한 다음, 당신의 시스템에 적합한 것을 선택한다. 언어 보충판(language supplements)가 필요하지 않다면, Informix Connect Version 7.24.UC5와 Informix SE Version 7.24.UC5만 다운로드 받으면 된다.

libc5 버전 `informix_se___esql_c_bundle_for_linux.zip`라는 이름의 9.5 Mb짜리 zip파일을 받으면 끝난다.

glibc 버전 `informix_se___esql_c_bundle_7_24_uc5_for_linux_on_kernel_2_0_34__glibc_2_0_7__red_hat_.zip`라는 이름의(장난아님!) 12.1Mb zip파일을 받으면 끝난다.

2.4 IDN에 등록하기

강력히 권장하지만, 완전히 선택사항이다. IDN은 인포믹스 개발자 네트워크 (Informix Developers Network)이다. IDN의 등록된 영역을 통해 당신은 인포믹스에 의해 운영되는 리눅스 사용자 포럼에 접근해서 기술 문서나 다른 정보들을 구할 수 있다.

2.5 IIUG에 등록하기

강력히 권장하지만, 이것도 역시 선택사항이다. IIUG는 국제 인포믹스 사용자 그룹 (International Informix Users Group)이다. 인포믹스가 그들의 제품을 리눅스로 포팅하도록 설득한 것에 대해 우리는 IIUG에 감사해야 한다. IIUG사이트의 중요한 부분들이 리눅스 사용자들을 위해서 Informix-SE를 당신의 시스템에 설치하고 동작시키기 위한 몇가지 HOWTO들과, 완전하게 검색 가능한 `comp.databases.informix` 의 아카이브, 그리고 당신이 새로 Informix-SE를 설치하고 유지하는 일들을 더 쉽게 만들어 줄 코드의 grundle을 포함하고 있다.

2.6 함정은 무엇인가?

개발자 꾸러미는 무료이고 만기가 없으므로, 당신 마음대로 사용할 수 있다. 말하자면 "함정은 없다." 나는 그것이 반드시 "함정"라고 생각하지는 않지만, 만일 당신이 Informix-SE상에서 어플리케이션을 개발하고 배포한다면 당신은 엔진의 각 사용자에 대해서 런타임 라이센스를 구입해야만 한다. 개발에 필요하다면 지원 계약을 맺을 수도 있다. 인포믹스는 이 일을 무료로 하지 않으며 그들의 노력에 대한 어느 정도의 대가를 기대한다. 더 자세한 정보는 `http://www.informix.com/cgi-bin/contact.pl` 를 통해 인포믹스의 영업 담당 부서에 연락할 것.

2.7 기술 지원이 가능한가요?

물론이다. 그러나 공짜를 기대하면 안된다(TANSTAAFL). 지원 계약을 맺으려면 비용을 지불해야 한다. 왜냐고? RDBMS 사업에 익숙한 사람들은 지원 계약이 RDBMS 벤더의 세가지 주요 수입원 중의 하나라는 것을 이해할 것이다(나머지 두 수입원은 런타임 라이센스와 컨설팅 비용이다). 이것은 다른 소프트웨어 벤더들의 "무료" "기술 지원"에 익숙한 모래 상자 속의 새로운 아이들에게는 놀라울 것이다. 그러나, 일반적으로 지원 계약은 마이크로소프트나 델(Dell)같은 회사들이 제공하는 무료 기술 지원보다 더 나은 서비스와 지원을 당신에 제공한다는 것은 말할 필요도 없다.

기술 지원 계약에 대한 더 자세한 정보는: `http://www.informix.com/informix/services/csp`를 참조 할 것.

`comp.databases.informix` 에 몇몇 인포믹스 직원들과 인포믹스 제품의 오랜 사용자들이 많이 살고 있다. 내 자신의 경험에서 볼 때 c.d.i와 그것의 동료 메일링 리스트는 지원의 신속한 출처이다. c.d.i.의 거주자들은 인내심이이 강하고, 꽤나 관대하지만, 게시하기(posting) 전에 DejaNews (`http://www.dejanews.com`) 를 검색하고 그룹 내에 숨어서 기다리면서 사전 조사할것을 강력히 권장한다. "c.d.i의 태양아래에 새로운 것은 아무것도 없다"(Ecclesiastes[역자주:구약성서중의 하나]에겐 미안하지만). 그러므로 당신은 David Williams이 관리되고 IIUG 사이트에서 운영하는 `InformixFAQ` 를 읽기를 원할 것이다.

제 3 절 설치와 설정

주: 인포믹스의 사이트에 있는 설치 지침과 IIUG사이트에 있는 Nils Myklebust의 멋진 HOWTO에는 몇가지 중복되는 부분들이 있다 (10(추가적인 리소스)를 볼것). 이것은 두가지 이유 때문에 사려 깊은일이다:

1. 바라건데, 반복을 통해, 우리는 몇가지 중요한 점들을 알게될 것이다.
2. 독자들은 아직 다른 문서에서 그것들을 발견하지 못했을 수도 있다.

3.1 설치 전 작업

아래의 예제는 다음을 가정한다:

- 호스트 기계의 이름은 *frodo*이고 도메인은 *hobbit.com*이다
- 당신은 root 사용자로 로긴되어 있다
- 당신은 Bourne shell 계열(Bourne, Korn, bash, pdksh)을 사용하고 있다

3.1.1 *informix*그룹 만들기

대부분의 리눅스 배포판은 `addgroup`이나 `groupadd` 둘 중의 하나를 갖고 있을 것이다. 이 프로그램들은 새로운 사용자 그룹을 만든다. 만일 당신의 시스템이 `usercfg`같은 GUI툴을 가지고 있다면 그것을 대신 사용해도 된다. 본질적인 정보만 바뀌지 않으면 된다.

```
<frodo> $ groupadd -g 777 informix
```

또는

```
<frodo> $ addgroup -g 777 informix
```

이 예제는 GID (group ID)가 777인 *informix*라는 이름의 그룹을 만든다. GID는 단지 예일 뿐이다; 당신은 아직 할당되지 않은 어떤 GID라도 사용할 수 있고 (할당된 GID를 볼려면 `cat /etc/group`), 특정한 값을 지정하지 않고 다음 가용한 GID를 사용할 수도 있다. 다음 단계에서 당신이 만드는 *informix* 사용자만이 *informix* 그룹의 멤버가 되어야 한다. 만일 다른 어떤 사용자가 이 그룹의 멤버라면, 그 사용자는 어떤 데이터베이스라도 변경하고 삭제할 수 있게 될것이다 - 좋은 생각이 아니다.

3.1.2 *informix* 사용자 계정 만들기

대부분의 리눅스 배포판은 `adduser`나 `useradd` 둘 중의 하나를 갖고 있다. 이 프로그램들은 새로운 사용자를 만든다. 만일 당신의 시스템이 `usercfg`같은 GUI 툴을 갖고 있다면, 그것을 대신 사용해도 된다. 본질적인 정보만 바뀌지 않으면 된다.

```
<frodo> $ useradd -u 777 -g informix informix
```

또는

```
<frodo> $ adduser informix
```

첫 번째 형식은 UID(-u 777)와 초기 그룹(-g informix)을 지정할 것을 요구한다. 그것은 초기 그룹이 *informix*이고 UID가 777인 *informix*사용자를 만든다. 이 UID는 단지 예일 뿐이다; 당신은 아직 할당되지 않은 어떤 UID라도 사용할 수 있고 (할당된 UID를 보기위해서는 `cat /etc/passwd`), 특정한 지정하지 않고 다음 가용한 UID를 사용할 수도 있다. *informix*사용자는 일차 그룹이 *informix*그룹이어야한다; 가능하다면, 이 사용자가 유일한 그룹원이어야 한다.

두 번째 형식은 `adduser`를 사용하는데, 대개는 (Caldera의 OpenLinux를 제외하고는 거의 항상) `useradd`프로그램의 대화식 셸 스크립트 프론트엔드이다. 만일 당신이 이 스크립트를 사용한다면 초기그룹을 지정하라고 할 때 `informix`를 입력한다. 전과정에 걸쳐 더 많은 제어권을 사용할 수 있는 `useradd`를 사용할 것을 권장한다. 자세한 것은 `man useradd`를 볼것.

Informix-SE에서는 *informix*사용자가 홈 디렉토리나 디폴트 셸을 가질 필요는 없다. 당신이 *informix*로긴할 경우는 좀처럼 없겠지만, 그렇게 한다고 해서 해가 될 것도 없다. 반면에 Informix Dynamic Server는 홈 디렉토리를 필요로 하므로, 나중에 리눅스용 IDS가 나올 때 그걸 설치할 계획이라면, 지금 홈 디렉토리를 만들어 두는 것이 그때 한 단계를 줄일 수 있을 것이다.

3.1.3 *dba*사용자 만들기

*dba*라는 이름의 사용자 계정을 3.1.2(<tt>informix</tt>사용자 계정 만들기)의 절차에 따라 만든다. 이 계정은 당신의 데이터베이스를 만들고 관리하는데 사용될 것이다. 여기에 두가지 규칙이 적용된다:

1. 데이터베이스를 만드는데 *informix*사용자를 사용하지 말것.
2. *dba*를 *informix*그룹의 멤버로 만들지 말것. 그렇게 하면 데이터베이스 엔진에 내장된 보안 메카니즘을 망쳐버릴 것이다.

이 사용자의 홈 디렉토리는 데이터베이스가 저장될 것이므로 그것의 위치에 조금 신경을 써야한다. 내 시스템의 경우에는, *dba*의 홈 디렉토리는 `/dbms`이고, 그것은 물리적으로 별도의 디스크에 위치하고 부팅할 때 루트 파일시스템에 마운트된다. 이렇게하면 데이터베이스에 피해 없이 루트 파일 시스템을 잃거나 그 거꾸로 되는 것이 허용된다.

3.1.4 설치 디렉토리 만들기

다음으로, 인포믹스를 설치할 디렉토리를 만들어야 한다. 을 보면 정적이고, 공유가능한 파일(문서, 라이브러리 또는 다른 바이너리들 같은)을 `/usr`이나 `/opt`에 설치할 것을 권장한다. 비 시스템 파일들은 종종 `/usr/local`에 설치된다. 내 시스템의 경우에는 그것들은 `/opt`에 있다. 당신은 당신이 좋아하는 어디라도 설치할 수 있을 것이다(`/tmp` 나 `/dev`은 권장되지 않음 ©).

```
<frodo> $ mkdir /opt/informix
<frodo> $ chown informix.informix /opt/informix
<frodo> $ chmod 755 /opt/informix
```

3.2 소프트웨어 설치하기

먼저 $INFORMIXDIR 환경 변수를 당신의 설치 디렉토리를 가리키도록 설정한다:

```
<frodo> $ export INFORMIXDIR=/opt/informix
```

다음으로, 패스를 설정한다. 그렇지 않으면 설치 스크립트가 제품표시 정보(branding information)를 찾지 못할 것이다:

```
<frodo> $ export PATH=$PATH:$INFORMIXDIR/bin
```

다운로드한 파일을 보관해둔 디렉토리로 가서 그것들을 unzip한다 (이 디렉토리는 설치 디렉토리와 다른 곳이어야 한다). 만일 당신이 CD-ROM을 갖고 있다면, 당신은 CD-ROM으로부터 바로 설치할 수 있다.

```
<frodo> $ cd /downloads
<frodo> $ unzip informix_se___esql_c_bundle_for_linux.zip
```

당신은 각각 ESQLC.TAR, SE.TAR와 ICONNECT.TAR가 들어있는 ESQLC, SE, ICONNECT 세 개의 디렉토리를 보게될 것이다. 당신의 설치디렉토리로 가서(이 예에서는 `/opt/informix`), tar파일들은 ESQLC, SE, ICONNECT 순서로 풀어야 한다.

[역자주: 현재 인포믹스는 기존의 ICONNECT, CLI등의 여러가지 클라이언트 연결 제품군을 하나로 묶어 Client SDK라는 형태로 공급하고 있다. 자세한 내용은 `http://www.informix.com`를 참조 할 것.]

3.2.1 ESQL/C

ESQL.TAR 풀기:

```
<frodo> $ tar xvf /downloads/ESQLC/ESQL.TAR
```

ESQL/C를 설치하고 라이센스를 적용한다 (Intraware로 부터 받은 일련 번호와 라이센스 키를 요구할 것이다):

```
<frodo> $ ./installesql
```

3.2.2 SE

SE.TAR 풀기:

```
<frodo> $ tar xvf /downloads/SE/SE.TAR
```

SE를 설치하고 라이센스를 적용한다 (Intraware로 부터 받은 일련 번호와 라이센스 키를 요구할 것이다):

```
<frodo> $ ./installse
```

3.2.3 ICONNECT

주: 이 단계는 선택적이다. Informix-SE는 Informix-CONNECT를 설치하지 않아도 잘 동작한다. SE와 ESQL/C는 둘다 필요한 연결방법(connectivity)을 내장하고 있다. IConnect는 당신이 SE와 ESQL/C가 설치되어 있지 않은 시스템에 설치된 ESQL/C 바이너리와 엔진을 연결할 수 있도록 하기 위한 것일 뿐이다.

ICONNECT.TAR 풀기:

```
<frodo> $ tar xvf /downloads/ICONNECT/ICONNECT.TAR
```

ICONNECT를 설치하고 라이센스를 적용한다 (Intraware로 부터 받은 일련 번호와 라이센스 키를 요구할 것이다):

```
<frodo> $ ./installconn
```

3.3 설치후 작업

설치후 작업은 거의 없다. 인포믹스에게 어떻게 실행되어야 하고, (선택적으로) 당신의 TCP/IP 설정을 조정하는 것이 전부다.

3.3.1 sqlhosts 파일 설정

우선, `$INFORMIXDIR/etc/sqlhosts`의 보관용 사본을 만들어 둘것. 당신이 완전히 anal이라면, 당신은 `cp -p` 또는 `cp -a` 를 사용하여 원래 파일의 시간과 날짜까지도 유지할 수 있다.

`sqlhosts`의 사본을 만든 다음, 원래 파일을 수정한다. 당신의 아래처럼 보이는 줄을 삭제하거나 주석처리(# 을 사용하여) 할 수 있다.

```
demo_on   onipcshm   on_hostname   on_servername
```

이 줄은 Informix Dynamic Server를 위한 것이므로, 리눅스에서는 아직까지 불가능하다. 호스트이름은 uname -n의 출력과 정확히 일치해야 하며, 그렇지 않으면 -25591 에러를 얻게 될것이다. 별명을 사용하는 것은 **동작하지 않는다**. 그 항목은 이렇게 보일 것이다

```
demo_se   seipcpip   frodo.hobbit.com   sqlexec
```

1. 항목 1 은 데이터베이스 서버의 이름이다.
2. 항목 2 는 통신 프로토콜이다(인포믹스의 문서에서는 종종 "nettype"으로 언급된다)
3. 항목 3 은 서버를 실행하고 있는 호스트의 이름이다.
4. 항목 4 는 서비스 이름이다(기술적으로는, *seipcpip* 연결에서는 서비스 이름은 사용될 엔진이나 바이너리를 가리킨다. 이 경우에는 sqlexec).

*seipcpip*프로토콜은 무명 파이프를 사용하고, 로컬 연결 프로토콜로만 사용될 수 있다. 나는 *seipcpip*를 사용하지 **않으려고** 억지를 부리는 이유를 정말 모르겠다. 특히 개발 환경에서는(그런데, 그것은 Informix-SE의 이번 릴리즈에 뒤쳐진 억측이다). 그래서, 내는 그 점에 대해 단언한다: 만일 그렇게 하지 않을 특정한 이유가 없다면, Informix-SE로 어떤 일을 하기 위해서는 *seipcpip* nettype을 사용하라.

[역자주: 이 부분은 번역이 아무래도 이상하군요. 정말 그 이유가 궁금하신 분은 원문을 보시기 바랍니다.]

3.4 설치 테스트 하기

이제 당신은 괴상한 준비물을 가지고 있고, 당신의 설치를 확인할 준비를 되었다. 만일 당신이 경험있는 인포믹스 사용자가 아니라면, 정규 사용자 계정을 사용하는 것이 가장 좋다.

dbtest같은 테스트 데이터베이스를 보관할 디렉토리를 만들고, 그 디렉토리로 cd한다. $INFORMIXDIR 환경 변수를 인포믹스가 설치된 디렉토리를 가리키도록 설정하고; 비슷하게, $INFORMIXSERVER 환경 변수를 당신이 sqlhosts 파일에 지정한 서버 이름으로 설정한다. 마지막으로 $INFORMIXDIR/bin 를 당신의 $PATH 에 추가한다:

```
<frodo> $ export INFORMIXDIR=/opt/informix
<frodo> $ export INFORMIXSERVER=demo_se
<frodo> $ export PATH=$PATH:$INFORMIXDIR/bin
```

dbaccessdemo7 명령은 현재 디렉토리에 stores7 견본 데이터베이스를 생성하고 몇가지 예제 쿼리들도 그 디렉토리에 복사할 것이다.

```
<frodo> $ dbaccessdemo7
```

dbaccess명령은 엔진을 위해 쓸만한 화면 기반 인터페이스를 시작하고, 그것으로 데이터베이스와 작업할 수 있다.

```
<frodo> $ dbaccess
```

메뉴에서 **Query-language**를 선택한 다음(간단히 **Q**를 입력한다) **Choose** (**C**)를 선택한다. 이것은 견본 데이터베이스내에 가용한 쿼리들의 목록을 보여줄 것이다. 커서 키를 사용하여 sel_로 시작하는 쿼리들 중 하나를 선택하고, 엔터를 두 번 누르면, 어떤 결과를 보게될 것이다. 이 모든 일들이 작동하면, 당신은 당신의 기계에서 실행되는 Informix-SE를 얻은 것이다. 축하한다!

제 4 절 인포믹스 실행하기

4.1 데이터베이스 엔진 시작하기

4.1.1 sqlexec 사용하기

Informix-SE를 시작하기 위해서, 당신은 최소한 설정된 $INFORMIXDIR 과 $INFORMIXSERVER 환경변수를 갖고 있고, $INFORMIXDIR/bin 이 당신의 경로에 첨가되어 있음을 확실히 해라. 그렇다면, 간단히

```
<frodo> $ dbaccess
```

를 입력하면 세션이 시작될 것이다. 엔터를 누르면 당신이 열 수 있는 데이터베이스의 목록이 나타나고, 화살표 키를 사용하여 당신이 원하는 데이터베이스를 선택하고 엔터를 누른다.

4.1.2 sqlexecd 사용하기

데이터베이스 호스트(데이터베이스가 활동하는 기계)에 대한 원격지 접근을 허용하지 않으려면, `sqlexecd`를 사용하지 마라. 그러나 만일 당신이 끝까지 고집한다면 계속 읽어라.

sqlexecd는 대몬으로 실행되기 때문에, 루트에 의해 시작되어야만 한다. 이것을 완수하기 위한 다양한 방법이 있다. 어떻게 결정하든 간에, `$INFORMIXDIR`이 설정되어야 하고 반드시 유효한 데이터베이스 서버 이름이(`$INFORMIXDIR/etc/sqlhosts`에 정의된 대로) sqlexecd의 인자로 지정되어야만 한다.

자세한 것은 5.1(어떻게 원격지 연결을 허용하는가?)을 볼 것. sqlexecd를 부팅할 때 시작할려면, 다음을 rc.local (당신의 시스템 어디엔가 위치해 있다)에 추가한다. 물론, 당신이 SysV init 스타일의 스크립트를 좋아한다면, 당신은 그 방법을 사용해도 된다. 무슨 얘긴지 모르겠다면, rc.local 접근법을 사용하면 된다.

```
export INFORMIXDIR=/인포믹스를/설치한/디렉토리
$INFORMIXDIR/lib/sqlexecd 데이터베이스서버이름
```

sqlexecd를 다른 어떤 시간에 시작하려면, 위에서처럼 동일한 명령을 실행한다. 편의상, `/etc/profile`내에 `$INFORMIXDIR` 과 `$INFORMIXSERVER`를 설정하는 것이 일반적이다. 많은 사이트들은, 특히 대규모 시설들은, 이것들과 다른 인포믹스-관련 환경 변수를 설정하기 위해 하나 이상의 전역 초기화 스크립트를 사용한다.

4.2 멈추기

dbaccess를 사용할때는, dbaccess를 종료하는 것이(정상적인 상황에서) 관련된 `sqlexec` 프로세스를 자동적으로 죽일 것이므로, 직접적으로 `sqlexec`를 죽일 필요가 없다. 그러나, 비 정상적인 상황에서, `sqlexec` 그 자체 보다는 `dbaccess`나 셸 또는 컴파일된 ESQL/C 프로그램 같은 `sqlexec`의 부모 프로세스를 죽일 것을 강력히 권장한다. 그렇지 않으면 당신의 데이터베이스가 손상될 수 있기 때문이다. 부모 프로세스를 죽이는 것은 `sqlexec`에게 시그널을 보내고 정상적으로 셧다운할 기회를 줄 것이다.

이것은 내가 당한 일을 되풀이하지 않겠다면 충분이 중요하다: 모든 다른 방법이 비참하게 실패하지 않는다면 절대 `sqlexec`프로세스를 죽이지 마라. 이렇게 함는 것은 특히 트랜잭션 로그를 사용하는 데이터베이스에서, 당신의 데이터베이스를 쓰레기로 만들 위험과 가능성을 올려준다. 트랜잭션 로깅이 작동중일 때, `sqlexec`를 죽이는 것은 잠재적으로 불완전한 트랜잭션을 야기하고 그것들은 roll-back 되지 않는다: 작동중인 어떤 락도 해제되지않고, `sqlexec`와 함께 죽기 때문에, 다른 프로세스들은 부분적으로 갱신된 데이터를 계속 조작하게 될 것이다. 당신은 **진정으로** 이렇게 하길 원치 않는다! 다른 대안이 없는 것이 확실할때까지는 절대 `sqlexec` 프로세스를 죽이지 마라.

반면, `sqlexecd`를 죽이는 것은 어느 정도 무해하다. 그것의 유일한 목적은 단지 들어오는 접속 요청를 위해 포트를 듣고 있다가, 그들의 요구를 서비스하기 위해 `sqlexec` 프로세스를 실행하고(fork off), 또 다른 요청을 듣는 것이다.

4.3 이 에러 코드들은 무엇인가?

Informix-SE 에러 코드를 인간이-읽을 수 있는 형태로 바꾸기 위해서는, `finderr` 프로그램을 사용한다 (다시, `$INFORMIXDIR/bin`가 경로에 포함되어 있는지 확실히 할것):

```
finderr <-num> 에러 -num 에 대한 에러 텍스트를 리턴한다.
finderr <num> 에러 -num에 대한 에러 텍스트를 리턴한다.
finderr <+num> 에러 num에 대한 에러 텍스트를 리턴한다.
```

인포믹스의 Answers OnLine 사이트 (10.2(인포믹스 문서)를 볼것)에서 구할 수 있는 *Informix Error Messages* 지침서에서도 에러 번호를 찾아 볼 수 있다.

제 5 절 문제가 있나요?

5.1 어떻게 원격지 연결을 허용하는가?

원격지 클라이언트를 위해 듣고 있다가, 연결을 위해 `sqlexec` 프로세스를 실행하는(fire off) `sqlexecd` 내의 버그 #97893 (리눅스에 국한된)와 #97889 (솔라리스에 국한된 것으로 간주되는)로 인해, sqlexec 세션이 종료될 때 프로세스 테이블에 좀비를 남긴다. (5.2(프로세스 테이블의 이 좀비들은 무엇인가?)와 5.3(그것들은 어떻게 방지하는가?)를 볼것). 이 경고를 듣고도 당신이 여전히 고집한다면, 이렇게 해라:

- `$INFORMIXDIR/etc/sqlhosts`를 `sqlexecd` 대몬을 사용하도록 조정한다
  1. *seipcpip*를 *sesoctcp*로 변경한다.
  2. `sqlexec`를 `sqlexecd`로 변경한다.
- 서버 기계의 `/etc/services`를 조정한다. 다음의 줄을 추가하라

  ```
  sqlexecd      1536/tcp
  ```

  항목은 파일 내의 아무곳에나 위치할 수 있고. 이미 사용중이 아니라면 1536 대신 다른 어떤 포트 번호를 사용할 수 있다.
- 그 포트를 듣고 원격지 클라이언트를 위한 `sqlexec` 세션을 시작하기 위해소는 `sqlexecd`를 실행한다.

  ```
  $INFORMIXDIR/lib/sqlexecd demo_se
  ```

  이 예에서는 데이터베이스 서버 이름을 *demo_se*로 가정한다.

5.2 프로세스 테이블의 이 좀비들은 무엇인가?

그것들은 인포믹스 식으로 말하면 "우리는 회초리를 맞았지만 계속 똑딱거린다!"이다. ;-) 진지하게, 그것들은 버그 #97893의 표출이다. 이 버그는 데이터베이스에 대한 원격지 접근을 제공하기 위해 소켓(*sesoctcp*)을 사용할 때 가장 빈번하게 발생하는 것 같다. 로컬 연결 상에서 무명 파이프(*seipcpip*)를 사용할때도 발생한다는 보고도 있다. 재미있는 것은 glibc 릴리즈에서 버그 #97893 가 수정되었지만, 새로운 버그 #101155 가 등장했다는 것이다: SEIPCPIP 연결 프로토콜(파이프)가 레드햇 5.1 플랫폼에서 동작하지 않는 것이다.

5.3 그것들을 어떻게 방지하는가?

Jonathan Leffler (`jleffler@informix.com`)가 회피책 `nozombie.c` 를 게시했는데, 그것은 nohup가 같은 방식으로 사용된다. 다음은 Jonathan의 코드와 그의 언급이다. 이것은 공식적인(인포믹스에 의해 입증된) 수정이 아니고, 모든 경우에 동작하는 것은 아니라는 보고도 있다는 것에 주의하라. YMMV.

```
설명은 꽤 간단하다 -- 만일 프로세스가 SIGCHLD를 무시한다면, 그것은 좀비 자식들을 쌓지
않는다. 프로그램은 SIGCHLD에 대해 SIG_IGN 시그널 처리 모드를 설정한 다음 인자로
주어진 것을 그대로 실행한다. 만일 그것이 sqlexecd에 발생한다면,
그것은 SIGCHLD 시그널을 무시하게 되고, 그래서 여기저기에 좀비를 남기지 않는다.
```

```
/*
@(#)File:           $RCSfile: nozombie.c,v $
@(#)Version:        $Revision: 1.1 $
@(#)Last changed:   $Date: 1998/08/20 21:24:40 $
@(#)Purpose:        Prevent process from accidentally creating zombies
@(#)Author:         J Leffler
@(#)Copyright:      (C) JLSS 1998
@(#)Product:        :PRODUCT:
*/
/*TABSTOP=4*/
#include <signal.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef lint
static const char rcs[] = "@(#)$Id: nozombie.c,v 1.1 1998/08/20 21:24:40 jleffler Exp $";
#endif
/*
** Exec program specified by arguments with SIGCHLD signals ignored.
** This ensures that unless the program re-enables the SIGCHLD signal
** handling, it does not leave zombies around, even if it doesn't
** clean up behind its children.  This works on POSIX.1 systems (such
** as Solaris 2.6 and Linux) pretty straight-forwardly.
**
** Motivation: the initial version of sqlexecd 7.24.UC1 on Linux
** caused problems with lots of zombies.
**
**     nozombie $INFORMIXDIR/lib/sqlexecd [service]
*/
int main(int argc, char **argv)
{
    signal(SIGCHLD, SIG_IGN);
    execv(argv[1], &argv[1]);
    fprintf(stderr, "Failed to execv() %s\n", argv[1]);
    return EXIT_FAILURE;
}
```

Jonathan의 코드로 해결이 안되면, 최근에 informix.idn.linux에 나타난 방법을 시도해 보라. 그것은 signal 함수가 작동하는 방식을 재작성함으로써 좀비 문제를 수정한다. 먼저, 다음과 같이 `signalfix.c` 를 만든다:

```
#include "signal.h"
#include <unistd.h>
#include <stdio.h>
void *signal(int signum,void (*handler)(int))
```

```
{
  struct sigaction sa;
  sa.sa_handler=handler;
  sa.sa_mask=SA_NOMASK;
  sa.sa_flags=SA_RESTART;
  sigaction(signum,&sa,(struct sigaction *)NULL);
}
```

다음으로, /usr/include/signal.h 를 복사하고, 시그널 함수를 주석처리 한다. 그런 다음, signalfix.c를 이런 식으로 컴파일한다:

```
$ gcc -fpic -shared signalfix.c -o libsig.so
```

마지막으로, sqlexecd 를 실행한다:

```
$ LD_PRELOAD=/root/sqlexecfix/libsig.so $INFORMIXDIR/lib/sqlexecd servername
```

5.4 DBACCESS가 xterm에서 세그먼테이션 폴트를 발생시킨다!

직접적인 문제는 dbaccess가 명백히 termcap/terminfo 항목들을 유지하는 정적인 버퍼를 할당하고 당신의 항목이 이 버퍼가 유지하는 것보다 길다는 것이다. 가까운 미래의 어느날 ©, 나는 이것을 인포믹스에 보고할 것이고, 다음 릴리즈 사이클내에 수정된 것을 얻길 바란다.

그 동안에 회피할 수 있는 방법은:

1. $TERM 환경 변수를 linux, vt220 또는 vt100같은 xterm이 아닌 다른 것으로 변경한다
2. termcap이나 terminfo내의 관련된 항목들을 수정한다(물론 보관용 사본을 만들어둔 후에). - dbaccess는 "ti"나 "te" 항목을 사용하지 않으므로, 그것들은 삭제될 수 있다. 이 방법은 잘 동작하지만, ti/te 항목을 실제로 사용하는 모든 xterm세션에 영향을 미친다.
3. xterm-dbaccess같이 xterm 항목을 그대로 복제하고, "ti"와 "te"항목을 삭제하고, 당신이 xterm 창에서 dbaccess를 실행할 때 $TERM환경 변수를 xterm-dbaccess로 설정한다.
4. http://www.informix.com/idn-secure/Linux/WebPages/termcap.html 에서 구할 수 있는 대체 termcap/terminfo 파일을 사용한다. 거기에 나열된 경고와 제언들에 주의할 것.

Roger Allen (rja@sis.rpslmc.edu)은 세 번째 선택을 권장하고 설명하기를

```
/etc/termcap내에 다른 이름으로 xterm 항목의 사본을 만들고
새로운 이름을 당신의 TERM 설정으로 사용하거나 현재 항목을 변경하라.
어떤 인포믹스 매뉴얼의 부록 어딘가에 인포믹스가 사용하는 필드들의 목록이 있다.
나는 보통 ti와 te를 필드를 삭제한다.
당신도 DB-Access보다는 다른 인포믹스 툴들을 더욱 위한 것이지만,
선 그리기 문자, 그리고 몇가지 기능 키들을 할수 있게 하는
특별한 인포믹스 항목을 추가할 수 있다.
```

5.5 내가 사용하려는 포트가 이미 사용 중이다!

다른 것을 사용해라. 인정받은 인포믹스 포트는 1546이지만, 그것이 사용하는 소켓이 다른 서비스에 사용되는 중이 아니라면 어떤 것이라도 상관없다.

5.6 NFS-마운트된 파일시스템을 사용할 수 있는가?

아니다. 당신의 데이터베이스를 보관할 파일시스템을 NFS를 통해 마운트하기 보다는,

1. 원격지 호스트에 sqlexecd가 실행중이어야 하고,
2. 원격지 데이터베이스에 네트웍을 사용해서 접근해야만 한다.

SE의 리눅스 이외의 버전은 이것을 바이너리 내에서 강요하지만, 리눅스의 경우에는 해당되지 않을지도 모른다. Jonathan Leffler의 충고에 따르면, "속임수를 쓰고자 한다면 문제를 당연한 일로 여겨라 – 데이터 손상 문제들" 이것이 바이너리에 의해 강요되든 그렇지 않든, 당신의 데이터베이스를 NFS 위에 마운트하는 것은 (적어도) 두가지 이유 때문에라도 좋지 않은 생각이다:

1. NFS가 느리기 때문에, 접근이 느려질 것이다.
2. 만일 NFS마운트를 잃게 된다면, 당신은 어이없게 될 것이고, 데이터베이스 파일들에 무슨 일이 생겼을지는 뻔하다.

제 6 절 다른 인터페이스들

SE를 위한 알려진 인터페이스의 목록은 10.5(관련된 소프트웨어)을 볼것.

6.1 ODBC 드라이버가 가용한가?

그렇다. 10.5(관련된 소프트웨어)를 볼것.

6.2 JDBC를 사용할 수 있는가?

그렇다. 인포믹스는 최근에 어떤 type IV JDBC 드라이버를 발표했다. 비록 리눅스 정식으로 플랫폼에 대해 테스트되지는 않았지만, type IV JDBC 드라이버는 SE의 리눅스 버전에서 동작하는 것처럼 보인다. YMMV. 10.5(관련된 소프트웨어)를 볼것.

6.3 perl 인터페이스가 있는가?

그렇다, DBD::Informix. 10.5(관련된 소프트웨어)를 볼것.

6.4 PHP는 어떤가?

PHP의 최근 버전 (3.0.3)은 인포믹스를 지원한다. 10.5(관련된 소프트웨어)를 볼것.

제 7 절 SCO 바이너리를 실행하기

7.1 리눅스 상에서 SCO 바이너리를 실행할 수 있는가?

그렇다, iBCS를 사용하면 가능하다. 비록 그것이 당신의 인포믹스의 라이센스에 위배되긴 하지만. IANAL. Marco Greco는 정확하게 이것을 하는 멋진 페이지를 갖고 있다. 10.4(관련된 웹사이트)를 볼것.

제 8 절 다른 인포믹스 제품들

8.1 4GL/D4GL/IDS/여기에_제품을_추가하세요 는 언제 가능해 질것인가?

리눅스용 D4GL은 인포믹스로부터 직접 주문될 수 있다. 어쨌든 4GL이 어떤 상태인지는 아직 명확하지는 않다.

최근 인포믹스는 IDS, 즉 Informix Dynamic Server를 1998년 4/4분기 중에 출시할 것이라고 발표했다. 이것은 굉장한 소식이다!

[역자주: 이 문서를 번역하는 시점에서 리눅스용 IDS 7.3 이 발표되었으며, 알짜RH5.2에서 멋지게 동작하는 것을 확인했으며, 알짜RH5.2를 구입하면 트라이얼! 버젼 CD가 포함되어 있다.]

IIUG의 지지 위원회는 출시할 제품에 대해서, 출시되는 시기와 리눅스용 인포믹스에 관련된 다양한 다른 토론을 인포믹스와 계속해서 작업하고 있다. 만일 당신이 이러한 결정들에 영향을 미치고 싶다면 인포믹스의 웹 사이트의 설문 조사를 채워라: `http://www.informix.com/idn-secure/Linux/WebPages/survey.html`

제 9 절 기타

9.1 SE와 IDS/ODS의 차이점은 무엇인가?

일반적으로, SE는 IDS보다 느리고, 더 적은 데이터 형을 지원하고, 그것의 더 큰 형제들의 많은 특성들을 부족하다. 구체적으로 얘기하면,Informix-SE는 다음의 차이점이 갖고 있다:

- BLOB(Binary Large Objects)을 지원하지 않음
- VARCHAR 데이터 형을 지원하지 않음
- TEXT 데이터 형을 지원하지 않음
- BYTE 데이터 형을 지원하지 않음
- ESQL/C가 멀티쓰레드가 아니다
- ESQL/C가 TP/XA를 지원하지 않음
- TERMINFO는 리눅스를 지원하지 않음

그러나 이것을 당신이 SE를 무시해야 된다는 의미라고 오해하지 마라. 왜냐하면 관리가 매우 단순하기 때문에, Informix-SE는 wash-and-wear RDBMS이다. 그것을 설치하고, 몇가지 설정 파일의 몇 줄을 편집하고, 그리고 당신은 데이터베이스 경쟁에서 벗어난다.

SE 계속할 것이다. 계속... 이 특성들은 전임 MIS 관리자가 없는 사이트들의 관심을 끈다. 또한 SE는 목이 부러질 것 같은 성능을 요구하지 않는 어플리케이션을 수행할 때 뛰어나다; 게으른 것은 아니만, 그것은 그것의 몇몇 경쟁자처럼 빠른 데몬은 아니다. 다시 말하지만, 트레이드오프는 강인함을 위한 성능과 쉬운 관리 및 유지보수이다.

9.2 현재 출시된 것은 무엇인가?

이 FAQ가 마지막으로 갱신될 때 까지, 가장 최근 릴리즈 레벨은 1998년 7월 13일 출시된 Informix-SE 7.24.UC5이다.그러나, glibc 릴리즈에 기초한 코드는 1998년 10월 14일 위해 발표되었다.

9.3 sqlexec와 sqlexecd의 차이점은 무엇인가?

`sqlexec`프로그램은 데이터베이스를 만들고 조작하는 데이터베이스 엔진이다. 데이터베이스에 접근하는 각각의 프로그램은 자신만의 `sqlexec`사본을 갖게 된다. (이것이 `sqlexec`를 죽이는 것이 왜 좋지 않은 생각인지를 설명해 준다). 호스트 OS는 여러개의 SE 프로세스간의 자원 관리를 위한 필요한 락킹을 처리한다. 반면에, `sqlexecd`는 결코 데이터베이스를 건드리지 않는다. 그것의 *존재의 이유*는 데이터베이스 접근을 위한 요청을 듣고(대개 원격지 어플리케이션/클라이언트로 부터 발생하는) 요청한 클라이언트와 통신하고, 요청된 대로 데이터베이스를 조작하기 위해 `sqlexec`프로세스를 시작한다. 이 관계는 시스템에 로긴하려고하는 기계를 기다리는 `rlogind`와, `rlogind`가 사용자와 통신하기위해 실행시키는 셸 간의 차이점과 유사하다.

9.4 Informix-SE는 "쓰레드에 안전"한가?

비록 이 질문이 좋은 것이지만 그것은 문제가 되지 않는다. SE는 단일 쓰레드이고, 한 번에 하나의 요구를 다루는 동기적인 어플리케이션이다. 그래서, 쓰레딩은 정말로 엔진이 어떻게 행동하는가에 대해 영향을 미치지 않는다.

9.5 Informix-SE가 다중-프로세서 기계에서 실행될 것인가?

그렇다, 왜냐하면 밑에 있는 OS가 MP문제를 처리할것이므로. 이에 반해서, 당신이 "Informix-SE가 다중 프로세서의 이점을 사용하는가?"를 묻는 것이라면 대답은 "아니오."이다. SE는 단일 쓰레드이고 다중 프로세서로부터 그 자체로는 이득이 없다.

9.6 ESQL/C "쓰레드에 안전"한가?

그렇기도 하고 아니기도 하다. libc5 기반 버전은 쓰레드에 안전하지 않다. 왜냐하면 libc5가 쓰레드에 안전하지 않고, 나중에 glibc/libc6가 나오게된 동기 중의 하나였다. 또한 Jonathan Leffler는 ESQL/C가 오직 DCE 쓰레드와만 동작한다는 점을 지적했는데, 리눅스(대부분의 경우)는 오직 POSIX 쓰레드만을 지원한다. 비록 이들 두 쓰레드의 구조적으로 비슷하지만, 그것들은 호환성이 없다. 누군가 나의 잘못을 지적할 때까지, 나는 glibc기반 버전은 쓰레드에 안전*하다*고 주장할 것이다. 왜냐하면 엔진을 컴파일하기위해 사용된 C 라이브러리가 쓰레드에 안전하기 때문이다.

9.7 어떻게 참여할 수 있는가?

9.7.1 문서를 작성하라

만일 당신이 리눅스용 인포믹스에 참여하고 싶다면, Tom Schaefer `netd@mindspring.com` 가 HOWTO들과 관련된 문서들을 작성하고 적절한 저장소에 배포하는 것을 조정한다. HOWTO에서 다음 주제에 대해 특히 참여가 필요하다:

- ODBC 연결성
- JDBC 연결성
- Perl, DBD 와 DBI

다른 인포믹스 제품이 발표됨에 따라, 우리는 그것들을 위한 HOWTO와 FAQ도 필요할 것이다.

9.7.2 베타 테스트하기

다른 제품들이 공식적인 출시가 가까워짐에 따라, 인포믹스는 베타 테스터가 필요할 것이다. Art Kagel이 베타 프로그램을 조정한다. 더 자세한 정보는, `kagel@erols.com` 에서 그와 접촉할 것. 최소한 다음의 정보를 포함하면 인포믹스는 테스트될 제품과 당신의 인포믹스 경험 정도를 당신이 테스트하게 될 제품의 플랫폼을 일치시킬 수 있다:

- 리눅스 배포본과 버젼
- 잠재적인 테스트 데이터베이스의 크기
- 테스트 어플리케이션의 복잡성과 유형
- 하드웨어의 종류(단일-, 이중- 혹은 다중-CPU)
- 테이프 저장 장치
- 일반적인 경험의 정도
- 당신이 테스트하길 원하는 특정 인포믹스 제품

물론, Non-Disclosure Agreement (NDA)에 서명할 필요가 있을 것이다.

9.7.3 IIUG에 가입하기

국제 인포믹스 사용자 그룹은 인포믹스가 그들 제품의 리눅스버젼을 발표하도록 설득하는데 조력이 되었다. 지지 위원회는 이 노력을 계속할 것이다. 현재, IIUG에 가입하는 것은 무료이며, 당신은 인포믹스에 영향을 미치고, 또한 인포믹스와 리눅스용 인포믹스를 사용하는 사람들과 교류할 수 있는 기회를 얻게된다. `http://www.iiug.org` 그들을 만날 수 있다.

9.7.4 Informix에게 메일 쓰기

당신은 항상 인포믹스에게 직접적으로 메일을 쓸 수 있고 그들에게 당신이 리눅스 환경으로 포팅되길 바라는 특정한 인포믹스 제품을 말할 수 있다. 그들은 듣고 있다.

9.8 "SE" 가 무슨 뜻인가?

Standard **E**ngine. Informix-SE는 cooked, 즉 포맷된, 파일 시스템에서 동작하는 있는 ISAM-기반 데이터베이스 엔진이다.

9.9 재미있는 사소한 일들

그러한 사소한 일들에 흥미가 있는 이들을 위해, Carlton Doe가 인포믹스의 발표 뒤 처음 며칠동안 Intraware사이트로부터의 다운로드가 초기에 대략 일일 700회를 기록했다고 보고했다. 그것들은 그 이후로 대략 일일 300정도로 떨어졌다. 인포믹스는 런타임 라이센스나 지원 계약의 판매에 대한 정보를 아직 발표하지 않았다. 마이크로소프트 윈도 NT를 위한 비슷한 프로그램이 리눅스 개발자 꾸러미와 같은 때에 발표되었다. 최소한 초기에는 리눅스 다운로드가 NT다운로드를 앞질렀다. ;-).

제 10 절 추가적인 리소스

10.1 Informix의 리눅스 웹 페이지

- http://www.informix.com/linux
- http://www.informix.com/solution/linux/freelx.html

10.2 Informix-SE에 대한 문서

- Answers OnLine http://www.informix.com/answers/oldsite/answers/english/se72xux.html

10.3 설치 지침(HOWTO)

- IIUG http://www.iiug.org/techinfo/linux/linux_top.html
- Informix (IDN 멤버쉽 필요) http://www.informix.com/idn-secure/Linux/WebPages/install.html
- glibc-기반 시스템에 libc 설치하기 http://sunsite.unc.edu/LDP/HOWTO/Glibc2-HOWTO.html
- 리눅스용 Informix-CLI 설치하고 사용하기 http://www.iiug.org/techinfo/linux/Howto_Informix_CLI.html

10.4 관련 웹사이트

- Linux상에서 SCO 바이너리 실행하기 http://www.ctonline.it/\~{}marcog/ifmxlinux.htm
- InxUtil http://www.inxutil.com
- 국제 인포믹스 사용자 그룹 http://www.iiug.org
- 4GL 컴파일러 http://www.aubury.demon.co.uk/a4gl.html
- 또 다른 4GL 컴파일러 http://www.querix.com
- 일본 인포믹스 http:/www.informix.com/jp

10.5 관련된 소프트웨어

- ODBC 드라이버
  - Intraware http://www.intraware.com/informix
  - OpenLink Software http://www.openlinksw.com
  - Informix ClientSDK www.intraware.com/informix/
  - SCO SQL-Retriever www.sco.com/vision/products/sql/retriever
- JDBC 드라이버
  - OpenLink Software http://www.openlinksw.com
  - Informix http://www.informix.com/informix/products/javaannc.html
- 펄(perl)인터페이스는 **DBD::Informix**이다. http://www.cpan.org/modules/by-category/07_Database_Interfaces/ 에서 구할 수 있다. DBI와 DBD::Informix가 필요하다면 인포믹스를 뒤져 볼 것.
- PHP 3.0.3는 인포믹스를 지원한다. http://www.php.net를 방문하라.

10.6 유즈넷

- `comp.databases.informix` (gatewayed to informix-list)
- `informix.idn.linux` (IDN 멤버십 필요)

10.7 메일링 리스트

일반적인 정보를 위해서는, `http://www.iiug.org/usenet/iiug\_forums.html`를 방문할 것.

- linux-informix (IIUG에의해 운영됨). `majordomo@iiug.org` 에게 본문이 "subscribe linux-informix"인 메시지를 보낼것.
- informix-list (IIUG에의해 운영됨). `majordomo@iiug.org` 에게 본문이 "subscribe informix-list"인 메시지를 보낼것.
- SuSE 는 그들의 고유한 인포믹스 리스트를 `http://www.suse.com/Mailinglists/suse-informix.info.html`에 갖고 있다.

10.8 Informix FAQ

- HTML `http://www.iiug.org/techinfo/faq/faq\_top.html`
- Text `ftp://ftp.iiug.org/pub/informix/faq`

10.9 나에게 메일을 주세요!

바꿔 말하면: 이 문서로 부터 잘못된 또는 빼먹은 정보나 링크를 발견하면 `kwall@uswest.net` 에게 메일을 보내라. 데이터베이스를 설정하고, 4GL코드를 디버깅하거나 오래-실행하는 쿼리를 최적화하는 것을 도와달라고 쓰고 요청**하지마라**. 나는 상당히 바쁘고 당신은 나에게 자문 비용을 지불하길 원치않는다. 만일 이 FAQ의 릴리즈 노트를 본 뒤라면, 인포믹스 FAQ와 IIUG의 아카이브를 찾고, 당신이 여전히 당신이 필요한 답변을 찾지 못했다면 `comp.databases.informix`에 게시한다. 나는 이 FAQ에 포함시키기 위해 그것이 리눅스와 인포믹스에 관련이 있다면 거의 어떤 것에 대한 기꺼이 참작할 것이다.

제 11 절 Credits

인포믹스는 그들은 마침내 리눅스를 진지하게 받아들이고 그들의 제품을 리눅스에 포팅하는 절차를 시작했으므로 목록의 수위를 차지한다.

IIUG의 Carlton Doe에게 특별히 감사한다. 그는 나에게 IIUG의 linux-informix 메일링 리스트의 발행되지 않은 아카이브들에 대한 접근을 제공했고 나는 여기서 사용된 몇가지 재료를 수집했다. 나는 또한 나에게 데이터베이스 보안을 둘러싼 문제와 *seipcpip* 대 *sesoctcp*를 명확하게 해준 Jonathan Leffler 와 Nils Myklebust의 참여에 감사한다. Marco Greco, Art Kagel 그리고 Tom Schaefer는 정신적인 지원을 제공했다.

Matt Welsh 와 Linuxdoc-SGML 패키지 의 유지책임자들에게 감사한다. Linuxdoc-SGML은 이 FAQ의 작성을 단순하게해 주었고 내가 하나의 원본으로 모든 가용한 몇가지 포맷으로 만들 수 있게 해 주었다.

비평과 정정에 대해 감사한다(알파벳 순으로): 익명자들, Medhi Afshar, Orlando Andico, Joe Archer, Mike Aubury, Sebastian Paul Avarvarei, Dave, Fernando Fernandez, Marco Greco, Art Kagel, Christian Kirch, Jonathan Leffler, Michael Perry, Dan Peters, Mike Reetz, George Tereshko, Billy Wheeler, Rob Zook