

Enterprise Java Linuxra HOGYAN

Greg Wilkins gregw@mortbay.org az eredeti Gary Meyertől gary@meyer.net származik. v0.2, 2001.11.07

Ez a dokumentum leírja, hogyan telepítsünk Enterprise Java környezetet Linuxra, amely tartalmaz egy Java Development Kitet, egy Web szervert, támogatja a Java szervleteket, adatbázis hozzáférést biztosít JDBC-n keresztül, és támogatja az Enterprise Java Beaneket (EJB).

Contents

1	Bevezetés	4
1.1	Háttér	4
1.2	Kinek szól?	4
1.3	Új verziók	4
1.4	Szerzői jog és licenc	4
1.5	Nyilatkozatok	4
1.6	Lehetséges jövőbeli részek	4
1.7	Egyéb források	5
1.8	Visszajelzés	5
1.9	Magyar fordítás	5
2	Hogyan telepítsük a Java Development Kitet	5
2.1	Blackdown JDK	5
2.1.1	Háttér	5
2.1.2	Letöltés	6
2.1.3	Telepítés	7
2.1.4	Környezeti beállítások	7
2.1.5	A telepítés ellenőrzése	8
2.1.6	További információk	8
2.2	IBM Java Developer Kit	8
2.2.1	Háttér	8
2.2.2	Letöltés	8
2.2.3	Telepítés	9
2.2.4	Környezeti beállítások	9
2.2.5	A telepítés ellenőrzése	10
2.2.6	További információk	10
2.3	Kaffe	11
2.3.1	Háttér	11

2.3.2	Letöltés és telepítés	11
2.3.3	Környezeti beállítások	11
2.3.4	A telepítés ellenőrzése	11
2.3.5	További információk	12
2.4	Sun J2SE	12
2.4.1	Háttér	12
2.4.2	Letöltés	12
2.4.3	Telepítés	12
2.4.4	Környezeti beállítások	13
2.4.5	A telepítés ellenőrzése	13
2.4.6	További információ	14
3	Hogyan telepítsünk Webszervert?	14
3.1	Apache	14
3.1.1	Háttér	14
3.1.2	Letöltés, telepítés és a környezeti beállítások	14
3.1.3	A telepítés ellenőrzése	15
3.2	IBM Domino	15
3.3	IBM HTTP Szerver	15
3.3.1	Háttér	15
3.3.2	Letöltés	15
3.3.3	Telepítés	16
3.3.4	Környezeti beállítások	16
3.3.5	A telepítés ellenőrzése	17
3.3.6	További információk	17
3.4	Jetty HTTP kiszolgáló és szervlet konténer	17
3.4.1	Háttér	17
3.4.2	Letöltés	18
3.4.3	Telepítés	18
4	Java szervlet támogatás telepítése	18
4.1	Allaire JRun	18
4.2	Apache Tomcat	19
4.2.1	Háttér	19
4.2.2	Letöltés	19
4.2.3	Telepítés	19
4.2.4	Környezeti beállítások	19
4.2.5	Telepítés ellenőrzése	20

4.2.6	További információk	20
4.3	BEA WebLogic	20
4.4	Enhydra	20
4.5	IBM WebSphere	20
4.6	Locomotive	21
4.7	Jetty	21
5	Hogyan telepíts Java Server Pages (JSP) támogatást	21
5.1	Apache Jakarta	21
5.2	Caucho Resin	21
5.3	Jetty	21
6	JDBC támogatás telepítése	21
6.1	IBM DB2	22
6.2	MiniSQL	22
6.3	MySQL	22
6.4	Oracle	22
6.5	PostgreSQL	22
6.5.1	Háttér	22
6.5.2	Letöltés és telepítés	22
6.5.3	A telepítés ellenőrzése	24
6.5.4	További információk	25
6.6	Sybase	25
6.6.1	Háttér	25
6.6.2	Letöltés	25
6.6.3	Telepítés	25
6.6.4	A telepítés ellenőrzése	26
6.6.5	További információk	27
7	Hogyan telepítsük az Enterprise Java Bean (EJB) támogatást	28
7.1	BEA WebLogic	28
7.2	EJBoss	28
7.2.1	Háttér	28
7.2.2	Letöltés	28
7.2.3	Telepítés	28
7.2.4	Környezeti beállítások	28
7.2.5	A telepítés ellenőrzése	29
7.3	Bullsoft JOnAS EJB	32

1 Bevezetés

1.1 Háttér

Ezt a dokumentumot 1999 januárjában kezdte el írni Gary Meyer (gary@meyer.net), néhány héttel azután, hogy különböző nyílt forráskódú és védett Enterprise Java terméket telepített Linuxra. Itt az "Enterprise Java" úgy értelmezendő, mint a Java Enterprise API-k használata.

A dokumentumhoz, néhány frissítést Greg Wilkins (gregw@mortbay.org) tett hozzá 2001 novemberében, mindazonáltal a dokumentum egyes részei nem aktuálisak.

1.2 Kinek szól?

Ez a HOGYAN azoknak a szoftvereknek próbál segítséget nyújtani, akik érdekeltek az Enterprise Java kipróbálásában, fejlesztésében vagy terjesztésében Linuxon. Meghatározott tudás vagy tapasztalat Linuxban vagy a Java területén feltételezett.

1.3 Új verziók

A dokumentum újabb verziói megtalálhatók a Linux Documentation Project honlapján: <http://tldp.org/HOWTO/Enterprise-Java-for-Linux-HOWTO.html>

1.4 Szerzői jog és licenc

Gary Meyer és Greg Wilkins ezen írását a szerzői jog védi. Másolásához hozzájárulás szükséges, a dokumentum sokszorosítása és módosítása a GNU szabad dokumentációs licencének 1.1-es verziójának feltételeihez kötött, vagy bármely későbbi verzióéhoz, amit a Szabad Programokért Alapítvány adott ki, állandó részek, fedőlap és hátoldal szövegek nélkül. A licenc másolata elérhető a <http://www.gnu.org/copyleft/fdl.html> weboldalon.

1.5 Nyilatkozatok

A dokumentum ajánlásai abban segítenek, hogy amilyen gyorsan csak lehet egy Enterprise Java környezetet állíthass be Linux alatt. Ezek a javaslatok nem termék ajánlások vagy jóváhagyások. Ahogy közelebről megismered a rendelkezésre álló lehetőségeket, képes leszel saját környezeteket készíteni, és megtalálni a legmegfelelőbb megoldást a saját igényeid számára.

Ennek a HOGYANNak szándéka az "Enterprise Java"-t mint a Java Enterprise API-t értelmezni. Ez a HOGYAN nem célozza meg a skálázhatóságot, rendelkezésre állást, menedzselhetőséget és egyéb hasonló szempontjait a programnak, amelyeket az "enterprise" kifejezéssel gyakran összekapcsolnak.

1.6 Lehetséges jövőbeli részek

Ez a HOGYAN az Enterprise Java legnépszerűbb részeire koncentrál. Az következők lehetnének részei ennek a HOGYANNak.

- Integrált fejlesztői környezetek (IDEs) amelyek támogatják az Enterprise Javát Linuxra
- Java Naming and Directory Interface (JNDI) támogatása
- Java Mail API (JMAPI) támogatása

- Java Transaction Service (JTS) támogatása
- Java Interface Definition Language (JIDL) támogatása
- Java Messaging Service (JMS) támogatása
- Common Object Request Broker Architecture (CORBA) támogatása

Érdekel egy rész megírása? Akkor vedd fel a kapcsolatot a szerzővel, Greg Meyerrel a (gary@meyer.net) címen.

1.7 Egyéb források

Az App-Serv Center (Alkalmazás-kiszolgáló Központ) honlapját itt találod meg: <http://www.app-serv.com/>.

Java Enterprise Nutshellhez David Flanagantól és másoktól a <http://www.oreilly.com/catalog/jentnut/> helyen.

1.8 Visszajelzés

Kérlek a szerző címére küldd el a hozzászólásod és esetleges javaslatod: Gary Meyer (gary@meyer.net).

1.9 Magyar fordítás

A magyar fordítást [Minkó Mihály](#) készítette (2003.02.27). A lektorálást [Erdész Szabolcs](#) végezte el (2003.03.20). Bármilyen fordítással kapcsolatos észrevételt a linuxhowto@sch.bme.hu címre küldjetelek. A dokumentum legfrissebb változata megtalálható a [Magyar Linux Dokumentációs Projekt](#) honlapján.

2 Hogyan telepítsük a Java Development Kitet

Van néhány a Java Development Kit-ek közül, ami Linuxra is elérhető:

- [2.1](#) (Blackdown JDK)
- [2.2](#) (IBM Java Developer Kit)
- [2.4](#) (Sun J2SE)
- [2.3](#) (Kaffe)

Ha csak egy JDK-t akarsz kipróbálni, akkor azt javaslom, hogy a Sun J2SE-vel kezd, ha csak nem ajánlanak mást, azok a szoftverek amiket használsz vagy esetleg használni akarsz a későbbiekben. Továbbá, ha nyílt forrású megvalósítás érdekel, akkor Kaffe-ra van szükséged.

2.1 Blackdown JDK

2.1.1 Háttér

A Blackdown JDK a Sun JDK, Linuxhoz írt változata. E dokumentum írásának az időpontjában a Blackdown JDK verziószáma Intel architektúrán, a JDK 1.2.2, PowerPC-n pedig a JDK 1.1.8.

1999 decemberében a Sun bejelentette a Java 2 Platform Standard Edition (J2SE) elérhetőségét a Linux operációs rendszerekre. A Sun ezen bejelentése alapvető hatással volt a Blackdown-ra, mert ez egy portolás. Egy sajtóközleményben a Sun azt állítja, hogy "Az e heti bejelentés nem lett volna lehetséges a Blackdown-nal való együttműködés, a világ programozói és lelkes segítői nélkül. Innen kezdve a Blackdown lett Linux Java technológia szállítója. Az évek folyamán ennek szentelt fáradozásai lefektették az alapjait a Java 2 Linux platformra történő kiadásának; az ő fáradozásuk nélkülözhetetlen volt a kiadás sikeréhez."

Továbbá a Sun kiadvány azzal folytatja, hogy " A Blackdown.org továbbra is egy értékes forrás a Java technológia Linux platformra történő alkalmazása során és tartalmazva a JDK 1.1.x verziókat."

2.1.2 Letöltés

A Blackdown JDK elérhető a <http://www.blackdown.org> oldalon.

A Blackdown honlapján válaszd a letöltést és egy tükrözést (van magyar is - a ford.).

Válaszd ki azt a JDK verziót, amelyiket akarsz. Ha más programok, amiket használni szeretnél nem futnak egy bizonyos verziószámú JDK-val, akkor a legfrissebbet ajánlom, ami a dokumentum írásának idején a JDK 1.2.2-es.

Válaszd ki a megfelelő architektúrát, amire telepíteni akarsz. Intel architektúrára válaszd az i386-ot.

Bármely verziót választhatod, de ha bizonyos programok, amiket használni akarsz nem támogatják a kiválasztott verziót, akkor azt ajánlom, hogy töltsd le a legfrissebb verziót, ami elérhető.

A Blackdown JDK-hoz, egy csomó fájlt találsz, különböző fájl és csomagformátumban. Mindazonáltal biztosnak kell lenned abban, hogy a Linux disztribúciódban lévő libc támogatja azt, amelyiket letöltöttél.

Az elérhető fájlok a következőket tartalmazzák:

- *jdk* - A Java Development Kit tartalmaz mindent, amire a Java fordításához, futtatásához, nyomonkövetéshez szükséged lehet. Nem tartalmaz nemzetközi karakter konvertereket.
- *jre* - A Java Runtime Environment amely tartalmazza a nemzetközi karakter konvertereket.
- *rt* - Egy egyszerű Java Runtime Environment, ami nem tartalmaz nemzetközi karakter konvertereket.
- *i18n* - A nemzetközi fontkészletek és a karakter konverterek JAR csomagjai.
- *native* - További bináris csomagok a valós, többszálú futtatáshoz.

Azt javaslom, hogy csak a *jdk for java development angol* verzióját töltsd le.

Amikor letöltötted a Blackdown fájlokat, választanod kell a *libc5* és a *glibc* között, mint a lehetséges *glibc* verziók között.

A *libc* opciók tartalmazzák:

- *libc5* - Az öregebb, de még használatos Linux *libc*, a *libc5*.
- *glibc* - Az új *libc*.

Ha a Linuxnak egy újabb disztribúcióját használod, akkor a legvalószínűbb, hogy lesz *glibc*. Azt javaslom kezdetben a *glibc*-vel próbálkozz.

2.1.3 Telepítés

Azt javasolom, telepítsd a fájlokat a /usr/local könyvtárba. Miután letöltötted a fájlt, futtasd a következő parancsokat:

```
mkdir /usr/local/blackdown
mv jdk* /usr/local/blackdown
```

Ha a tarball formátumot töltötted le, akkor futtasd a következő parancsot:

```
tar zxvf [filename].tar.gz
```

Ahol a [filename] az állomány neve.

A /usr/local/blackdown könyvtárban látni fogsz egy olyasmi könyvtárat, hogy jdk1.2.2

A fenti egy JDK1.2.2 release candidate 3 intel architektúra példáját mutatja. Helyettesítsd a fájlnevet, verziószámot, kiadást és az architektúrát a megfelelőre. Minden disztribúciós csomagfájlt a fenti módszerrel nyiss meg.

2.1.4 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- JAVA_HOME
- PATH
- CLASSPATH

A JAVA_HOME környezeti változó arra a könyvtárra mutat, ahová a JDK csomagot telepítetted. Állítsd be a JAVA_HOME környezeti változót arra a könyvtárra, amelyikbe a Blackdown JDK csomagot telepítetted.

```
export JAVA_HOME=/usr/local/blackdown/jdk1.2.2
```

A \$JAVA_HOME/bin könyvtárban van a Java compiler (Java fordító; javac) és a Java Virtual Machine (Java Virtuális Gép; java) valamint egyéb, a fejlesztő környezethez szükséges programok.

Add hozzá a PATH változóhoz a \$JAVA_HOME/bin könyvtárat.:

```
export PATH=$JAVA_HOME/bin:$PATH
```

Ellenőrizd le, hogy a \$JAVA_HOME/bin a PATH elejéhez hozzá lett adva, így az általad telepített JDK csomagot fogod használni, nem pedig azt a JDK csomagot, ami a Linuxszal lett feltéve. A PATH változó ellenőrzéséhez kérdezd le az alapértelmezett Java fordító és JVM (Java Virtual Machine) nevét:

```
which javac
which java
```

A válasznak a \$JAVA_HOME/bin könyvtárban lévő javac és java csomagokkal összhangban kell lennie.

A CLASSPATH környezeti változó mutat minden JAR-ra és azokra a könyvtárakra, amelyekre a Java programok fordításához és futtatásához szükséged lesz.

A JDK1.2.2-es verziójához kezdetben nem szükséges JAR-okat adni a CLASSPATH változóhoz. A JAR-ok csomagolása lehet jar vagy zip fájl is.

```
export CLASSPATH=$CLASSPATH:.
```

2.1.5 A telepítés ellenőrzése

Most már futtathatsz és fordíthatsz egy egyszerű alkalmazást. Írd meg a következő programot (egy akármilyen szövegszerkesztővel, csak a fájl kiterjesztése "java" legyen - a ford.):

```
class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Fordítsd le a programot a Java fordítóval.

```
javac HelloWorld.java
```

Ha a fordító hibát jelez, akkor alaposan nézd át a szintaxist, valamint nézd meg a PATH és CLASSPATH beállításokat, hogy jók-e.

Futtasd a programot a JVM-mel.

```
java HelloWorld
```

Ha a JVM hibát produkál, akkor nézd meg a PATH és CLASSPATH beállításokat.

A következő eredményt kell látnod a program helyes futása esetén:

```
Hello, World!
```

Gratulálok, sikeresen telepítetted, beállítottad és tesztelted a Blackdown JDK csomagot Linuxon!

2.1.6 További információk

A BlackDown JDK csomagról további információt találsz a <http://www.blackdown.org> honlapon. Itt elérhető egy kiváló FAQ.

2.2 IBM Java Developer Kit

2.2.1 Háttér

Az IBM Java fejlesztői környezet és futtatási környezet kompatibilis a Sun Javával és tartalmazza a legfrissebb karbantartott eszközöket. (Az IBM weboldaláról.)

Ezen dokumentáció írásakor az IBM Java fejlesztői környezet a JDK 1.1.8 verziója aktuális, ez csak Intel architektúrára telepíthető.

2.2.2 Letöltés

Az IBM Java Developer Kit letölthető a <http://www.ibm.com/java/jdk/118/linux> címről.

A letöltéshez először regisztrálnod kell magad az IBM weboldalán, és el kell fogadni a szerződés feltételeit.

Az letölthető fájlok tartalmazzák:

- *ibm-jdk-1118-linux-x86.tgz* - The Java Development Kit; mindent tartalmaz ami egy Java program fordításához, futtatásához és nyomkövetéséhez szükséges.
- *ibm-jre-1118-linux-x86.tgz* - The Java Runtime Environment; a Java programok futtatásához szükséges fájlokat tartalmazza.

Mivel fejleszteni akarsz javasolom, hogy a IBM-JDK csomagot töltsd le. :)

2.2.3 Telepítés

Azt javasolom, hogy a /usr/local könyvtárba telepítsd a fájlokat. A fájlok letöltése után futtasd a következő parancsot:

```
mkdir /usr/local/ibm
mv ibm-jdk-1118-linux-x86.tgz /usr/local/ibm
```

Most kicsomagolhatod a disztribúciós csomagot. Ezt a következő paranccsal teheted meg:

```
tar zxvf ibm-jdk-1118-linux-x86.tgz
```

A /usr/local/ibm könyvtárban most már láthatod a jdk.118 könyvtárat.

A fent említett példák Intel architektúrán nyújtanak segítséget.

Helyettesítsd a fájlneveket a megfelelőkkel.

2.2.4 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- JAVA_HOME
- PATH
- CLASSPATH

A JAVA_HOME környezeti változó utal a telepített JDK csomagod home könyvtárára. Állítsd be a JAVA_HOME változót arra a könyvtárra, amelyikbe telepítetted az IBM Java fejlesztői készletet.

```
export JAVA_HOME=/usr/local/ibm/jdk118
```

A \$JAVA_HOME/bin könyvtárban van a Java compiler (Java fordító; javac) és a Java Virtual Machine (Java Virtuális Gép; java) és egyéb, a fejlesztéshez nélkülözhetetlen programok. Add hozzá a \$JAVA_HOME/bin könyvtárat a PATH változóhoz.

```
export PATH=$JAVA_HOME/bin:$PATH
```

Ellenőrizd, hogy a \$JAVA_HOME/bin könyvtárat hozzáadtad a PATH elejéhez, így az telepített JDK csomagot fogja használni a géped, nem pedig azt, ami a Linux disztribúcióval jött.

A PATH változó ellenőrzéséhez kérdezd le az alapértelmezett Java fordító és JVM (Java Virtual Machine) nevét:

```
which javac
which java
```

A válaszból kiderül melyik javac és java van a \$JAVA_HOME/bin könyvtárban. A CLASSPATH környezeti változó utal a JAR-okra és a könyvtárakra, amelyekre szükséged lesz ahhoz, hogy Java programokat fordíts és futtass.

Javasolom, add hozzá a következő JAR-okat a CLASSPATH változóhoz. Ezek különböző .jar vagy .zip fájlokba vannak csomagolva.

Például:

```
export CLASSPATH=$JAVA_HOME/lib/classes.zip
export CLASSPATH=$CLASSPATH:.
```

2.2.5 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egy egyszerű programot. Készítsd el a következő programot:

```
class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Fordítsd a programot a Java fordítóval:

```
javac HelloWorld.java
```

Ha a fordító hibát jelez, akkor nézd meg alaposan a szintaktikát és ellenőrizd a PATH és a CLASSPATH változó beállításait.

Futtasd a programot JVM-mel.

```
java HelloWorld
```

Ha a JVM hibát jelez ellenőrizd a PATH és a CLASSPATH beállításaid.

Egyébként a következő kimenetet kell látnod:

```
Hello, World!
```

Gratulálok, telepítetted és tesztelted az IBM Java Developer Kit csomagot Linuxon!

2.2.6 További információk

További információt találsz az IBM Java weboldalán a <http://www.ibm.com/java> címen.

2.3 Kaffe

2.3.1 Háttér

A Kaffe egy nyílt forráskódú implementációja a (Java Virtual Machine) Java Virtuális Gépnek és osztálykönyvtáraknak. Ezen dokumentum készítésének idején a Kaffe "leginkább a JDK 1.1-el működött együtt, néhány részt kivéve" és "egyres részei már JDK 1.2 (Java2) kompatibilisek". (Idézet a Kaffe weboldaláról.)

Előfordulhat, hogy a Kaffe már a Linux disztribúciódban benne van, hiszen a Kaffe is nyílt forráskódú.

2.3.2 Letöltés és telepítés

Azt javaslom, hogy inkább azt a Kaffe csomagot használd, ami a Linux disztribúcióddal érkezett, mintsem ami letölthető a Kaffe honlapról.

Alternatívaként a Kaffe megtalálható a <http://www.kaffe.org> címen.

A Kaffe weboldalán válaszd a legfrissebb kiadást. Ezen dokumentum írása idején ez az 1.0.5-ös verzió volt. A Kaffe verziószáma nincs kapcsolatban a JDK egyéb verziószámaival.

2.3.3 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- PATH
- CLASSPATH

A PATH változó ellenőrzéséhez kérdezd le az alapértelmezett Java fordító és JVM (Java Virtual Machine) nevét:

```
which javac
which java
```

A CLASSPATH környezeti változó mutatja, hogy hol találhatóak a JAR-ok és mely könyvtárakra lesz szükséged, hogy fordíts és futtass Java programokat.

Azt javaslom, hogy a JAR csomagokat add hozzá a CLASSPATH változóhoz. A JAR-ok .jar vagy .zip fájlalba lehetnek csomagolva:

Például:

```
export CLASSPATH=/usr/local/share/kaffe/Klasses.zip
export CLASSPATH=$CLASSPATH:.
```

2.3.4 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egyszerűbb alkalmazásokat. Készítsd el a következő programot:

```
class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Fordítsd le a programot a Java fordítóval:

```
javac HelloWorld.java
```

Ha a fordító hibát jelez, alaposan ellenőrizd le a szintaxist és ellenőrizd a PATH és CLASSPATH beállításaid.

Futtasd a programot a JVM-mel.

```
java HelloWorld
```

Ha a JVM hibát jelez ellenőrizd a PATH és a CLASSPATH beállításait.

A következő üzenetet kell látnod:

```
Hello, World!
```

Gratulálok, telepítetted, beállítottad és tesztelted a Kaffe csomagot Linux alatt!

2.3.5 További információk

További információt találsz a Kaffe-ről a <http://www.kaffe.org> honlapon.

2.4 Sun J2SE

2.4.1 Háttér

A Sun Java 2 Standard Edition (J2SE) a Sun Java 2 platformja Linux operációs rendszerekre. A dokumentum készítésekor a legfrissebb verzió a JDK 1.2.2, Intel architektúrára.

2.4.2 Letöltés

A J2SE letölthető a <http://developer.java.sun.com/developer/earlyAccess/j2sdk122> oldalról.

Regisztrálnod kell magad a Sunnál és el kell fogadnod a licencet, mielőtt letöltöd a programot.

2.4.3 Telepítés

Azt javasolom telepítsd a fájlokat a /usr/local könyvtárba. Miután letöltötted a fájlokat futtasd a következő parancsokat:

```
mkdir /usr/local/sun
mv jdk1_2_2rc1-linux-i386.tar.gz /usr/local/sun
```

Most kicsomagolhatod a disztribúciós csomagot. Futtasd a következő parancsot:

```
tar zxvf jdk1_2_2rc1-linux-i386.tar.gz
```

Az /usr/local/sun könyvtárban most már láthatod a jdk1.2.2 könyvtárát.

Az alábbi példák a JDK 1.2.2 verzióra, Intel architektúrára készültek. Helyettesítsd a fájlneveket, ahol szükséges.

2.4.4 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- JAVA_HOME
- PATH
- CLASSPATH

A JAVA_HOME arra a könyvtárra mutat, ahol a JDK csomagod van. Állítsd be a JAVA_HOME változót arra a könyvtárra, ahová a J2SE programot telepítetted:

```
export JAVA_HOME=/usr/local/sun/jdk1.2.2
```

A \$JAVA_HOME/bin könyvtárban van a Java compiler (Java fordító; javac) és a Java Virtual Machine (Jáva Virtuális Gép; java) és egyéb, a fejlesztéshez szükséges programok. Add hozzá a \$JAVA_HOME/bin könyvtárat a PATH változóhoz.

```
export PATH=$JAVA_HOME/bin:$PATH
```

Ellenőrizd, hogy a \$JAVA_HOME/bin könyvtárat hozzáadtad a PATH elejéhez, s így a telepített JDK verziót fogod használni, nem pedig azt, amelyik a disztribúcióddal jött.

A PATH változó ellenőrzéséhez kérdezd le az alapértelmezett Java fordító és JVM (Java Virtual Machine) nevét:

```
which javac
which java
```

Az eredményeknek a \$JAVA_HOME/bin könyvtárban található javac és java verzióknak kell lenniük.

A CLASSPATH környezeti változó azokra a JAR-okra és könyvtárakra mutat, amelyekre szükséged lesz a Java programok fordításához és futtatásához.

A JDK 1.2.2-höz, alapesetben nem kell JAR-okat adni a CLASSPATH változóhoz.

A JAR-ok .jar és .zip fájlokba lehetnek csomagolva.

```
export CLASSPATH=$CLASSPATH:.
```

2.4.5 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egy egyszerű alkalmazást. Készítsd el a következő kis programot:

```
class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Fordítsd le a programot a Java fordítóval.

```
javac HelloWorld.java
```

Ha a fordító hibát produkál, ellenőrizd le alaposan a szintaxist, és ellenőrizd a PATH és CLASSPATH beállításait.

Futtasd a programot a JVM-mel.

```
java HelloWorld
```

Ha a JVM hibát jelez ellenőrizd a PATH és a CLASSPATH beállításait.

A következő eredményt kell látnod a helyes futás esetén:

```
Hello, World!
```

Gratulálok, telepítetted, beállítottad és tesztelted a Sun J2SE csomagot Linux alatt.

2.4.6 További információ

További információt találsz a Sun J2SE javáról a <http://java.sun.com> honlapon. Itt kiváló vitafórumokat találhatsz, ahol valószínűleg választ kapsz a kérdéseidre.

3 Hogyan telepítsünk Webszervert?

Számos Webszerver található Linuxra. Például:

- [3.1](#) (Apache)
- [3.2](#) (IBM Domino)
- [3.3](#) (IBM HTTP Server)
- [3.4](#) (Jetty HTTP Server)

Ha csak egy Webszervert akarsz kipróbálni, azt javasolom, hogy kezd az Apache-al. Elsősorban azért, mert ez benne van a legtöbb Linux disztribúcióban, előre telepítve és talán már fut a rendszereden.

3.1 Apache

3.1.1 Háttér

Az Apache a legelterjedtebb HTTP szerver az Interneten. "Eredetileg az NCSA httpd démonon alapult, de azóta teljesen újra lett írva. Az Apache nyílt forráskódú". (Idézet az Apache weboldaláról.)

3.1.2 Letöltés, telepítés és a környezeti beállítások

Azt javasolom, hogy azt az Apache csomagot használd, amelyik a Linux disztribúcióddal érkezett.

Esetleg letöltheted a <http://www.apache.org> webhelyről.

3.1.3 A telepítés ellenőrzése

Az Apache működőképességének ellenőrzéséhez indítsd el a böngésződ, és írd be a következőt: "http://127.0.0.1". (ez a localhost IP címe).

Egy "It Worked!" feliratú weboldalt kell látnod, ha működik.

Ha ez nem így van, akkor ellenőrizheted úgy is, hogy beírod a következőt a RedHat Package Manager (RPM) alapú disztribúcióba:

```
rpm -q | grep apache
```

Az Apache elindításához, futtasd:

```
cd /etc/rc.d/init.d  
./httpd start
```

Megjegyzés: A httpd szkript helye, bootoláskor lehet, hogy máshol van a különböző Linux disztribúcióknál. További segítségért látogass el az Apache weboldalára és olvasd az Apache FAQ leírást a <http://www.apache.org/docs/misc/FAQ.html> honlapon.

3.2 IBM Domino

Ez a rész készülõben van.

További információkért keresd fel a <http://www.lotus.com/dominolinux> honlapot.

3.3 IBM HTTP Szerver

3.3.1 Háttér

Az IBM HTTP kiszolgáló, egy Apache IBM színekbe öltöztetve. Az IBM http kiszolgálóját mindenképpen meg kell ismerned, ha IBM WebSphere-el akarsz dolgozni.

A doksi készülésekor a naprakész verzió az 1.3.6.1-es.

3.3.2 Letöltés

Az IBM HTTP kiszolgáló letölthető a <http://www-4.ibm.com/software/webservers/htpservers/download.html> címről.

A letöltésre klikkelve válaszd az 56, vagy 128 bites ssl kódolást.

Regisztrálnod kell magad az IBM-nél, ki kell töltened egy marketing kérdőívet és el kell fogadnod a licencet, mielőtt letöltöd a programot. Az IBM http kiszolgáló futtatásához a glibc 2.0-ás vagy 2.1-es verziójára lesz szükséged. A glib az új Linux libc. Ha régebbi disztribúciód van, ami libc5-ön alapul, akkor nem fogod tudni használni az IBM http kiszolgálót.

Egy RedHat Package Manager (RPM) alapú Linux disztribúción futtasd a következő parancsot:

```
rpm -qa | grep libc
```

Hasonló eredményt fogsz látni:

```
glibc-2.1.2-11  
libc-5.3.12-31
```

Ebből tudható meg, hogy melyik libc5 és glibc verzió van a gépeden. A következő példában én mindkettőt telepítettem a rendszeremre. A glibc verzióm 2.1, tehát a glibc2.1-es verzióhoz tartozó csomagot kell letöltenem.

Javasolom, hogy töltsd le az összes tar fájlt a Linuxod glib verziójához, mivel viszonylag kis méretűek. Akármennyit is töltesz le, a szerver fájlra szükséged lesz. RedHat 6.0 és az ebből származó disztribúciókhoz csak a RedHat 6.0-ra lesz szükséged. A glibc 2.0-ra alapuló verziókhoz szükséged lesz a libstdc fájlra is.

3.3.3 Telepítés

Javasolom a fájlokat az /usr/local könyvtárba telepítsd. A letöltés után add ki a következő parancsokat:

```
mkdir /usr/local/ibm  
mv HTTPServer.linux.* /usr/local/ibm
```

Most már kicsomagolhatod a tar fájlt:

```
tar xvf [filename].tar
```

Ahol a [filename] az állomány neve.

Az /usr/local/ibm könyvtár alatt most már látnod kell egy IHS könyvtárat.

Használd a RedHat Package Manager programot (RPM) az rpm fájlok telepítéséhez, amiket a tar fájl tartalmazott. Ha szükséged van a libstdc fájlra, akkor először azt telepítsd. Azután telepítheted a szerver rpm fájlját, valahogy így:

```
cd IHS  
rpm -i IBM_HTTP_Server-1.3.6-2.i386.rpm
```

A fenti példák 1.3.6-os verzióra, Intel architektúrára készültek. Helyettesítsd a fájlneveket, ahol szükséges. A telepítés után könnyen törölheted az rpm fájlokat.

3.3.4 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- PATH

Az IBM http kiszolgáló az /opt/IBMHTTPServer könyvtárba telepíti magát. Ezt az útvonalat hozzá kell adnod a PATH változóhoz.

```
export PATH=/opt/IBMHTTPServer/bin:$PATH
```

Ellenőrizd, hogy az /opt/IBMHTTPServer/bin útvonalat hozzáadtad a PATH változóhoz, így az a Webkiszolgáló indul el, amit telepítettél, nem pedig az amelyik a Linux disztribúcióddal érkezett.

A PATH változó beállításának ellenőrzéséhez nézd meg melyik Apache kontrollert használod:


```
which apachectl
```

A válasznak az `/opt/IBMHTTPServer/bin` könyvtárban lévő `apachectl`-re kell mutatnia.

Figyelem: mivel az IBM http kiszolgáló az Apache-on alapul, ezért az Apache controllert használja induláskor és leálláskor. Ha az IBM http kiszolgálót és az Apache kiszolgálót is telepítetted a gépedre, akkor különösen figyelj a `PATH` változóra, hogy a megfelelő kiszolgálót használd!

Lehet, hogy módosítanod kell az IBM http kiszolgáló konfigurációs fájlját. A konfigurációs beállítások az `/opt/IBMHTTPServer/conf/httpd.conf` fájlban vannak. A két legfontosabb változtatás amire szükség lehet, az a kiszolgáló nevének és a portnak a beállítása. Keresd a "ServerName" és a "Listen" kulcsszavakat a `http.conf` fájlban. A kiszolgáló nevének beállíthatod a gépedet vagy az IP címed. Ha a számítógéped DHCP szolgáltatást használ az IP cím beállításához, akkor a "hostname" megadása jobb választás lehet.

A számítógép nevének használatához a "hostname"-nek és az IP címnek regisztrálva kell lennie a DNS-ben.

Azonfelül, ha más web-kiszolgálót is futtatsz a gépen, akkor az IBM http kiszolgálónak egy másik portot kell adnod, hogy mindkettő párhuzamosan és megfelelően működjön.

Íme egy példa a `httpd.conf` fájl beállításához:

```
ServerName 192.168.0.4
Listen 3000
```

3.3.5 A telepítés ellenőrzése

Az IBM http kiszolgáló indításához írd be a következőt:

```
/opt/IBMHTTPServer/bin/apachectl start
```

Az IBM http kiszolgáló rendesen működésének ellenőrzéséhez indítsd el a böngésződ és írd be a következő URL-t: `http://192.168.0.4:3000` (helyettesítsd a megfelelő IP címmel és port számmal, amiket a saját `httpd.conf` fájlodba írtál.)

Egy "Welcome to the IBM HTTP Server" weboldalt kell látnod. Gratulálok, telepítetted, beállítottad és tesztelted az IBM http kiszolgálót Linux alatt!

3.3.6 További információk

További információért keresd fel az IBM http kiszolgáló <http://www-4.ibm.com/software/webservers/htpservers/support.html> honlapját.

3.4 Jetty HTTP kiszolgáló és szervlet konténer

3.4.1 Háttér

A Jetty egy nyílt forráskódú, 100%-ban Javában írt http szervlet szervert. Egy teljes http/1.1 és szervlet konténer egyszerre. Egyszerű kezelhetőség, nagy hatékonyság, beágyazhatóság, kiterjeszthetőség és rugalmasság jellemzi, így ideális platform dinamikus http kérések kiszolgálására bármilyen Java program számára.

A Jetty használható önmagában http kiszolgálóként és Servlet konténerként, vagy beágyazható egy másik Java alkalmazásba (pl: a JBoss EJB konténer a Jetty-t használja, mint preferált kiszolgálót és konténert.)

Mint kombinált kiszolgáló és szervlet konténer, mindkét funkció hatékonyan fut egy Unix folyamatban. A telepítése és a beállítása olyan egyszerű, mint egy kis alkalmazásé.

3.4.2 Letöltés

A Jetty http kiszolgáló és Servlet konténer letölthető a

<http://jetty.mortbay.org> honlapról.

A Jetty az artistic liszenc alá tartozik, a teljes kódot tartalmazza és kereskedelemben terjeszthető.

3.4.3 Telepítés

A csomag egy gzipelt tar fájlban van, ami kicsomagolható a következő paranccsal:

```
gunzip < Jetty-x.x.x.tgz | tar xf -
```

Ez létrehoz egy Jetty-x.x.x. könyvtárat, ahol is az x.x.x a verziószámot jelenti.

A demó kiszolgáló futtatásához használd a következő parancsokat:

```
export JETTY_HOME=<jetty install directory>
export JAVA_HOME=<JRE install directory>
$JETTY_HOME/bin/jetty.sh run
```

A Jetty demójának indításához írd be a böngészőbe: <http://localhost:8080> .

A Jetty, JMX vagy JBoss környezet részeként is telepíthető. További információkért látogass el a <http://jetty.mortbay.org> honlapra, vagy olvasd el a README.TXT fájlt.

4 Java szervlet támogatás telepítése

Van néhány WEB kiszolgáló-bővítmény és alkalmazás-kiszolgáló Linuxon, amely támogatja a szervleteket. Ilyenek a következők:

- [4.1](#) (Allaire JRun)
- [4.2](#) (Apache Tomcat)
- [4.3](#) (BEA WebLogic)
- [4.4](#) (Enhydra)
- [4.6](#) (Locomotive)
- [4.5](#) (IBM Websphere)
- [4.7](#) (Jetty)

4.1 Allaire JRun

A leírás elkészítése folyamatban van.

További információk a <http://www.allaire.com/products/jrun/> oldalon.

4.2 Apache Tomcat

4.2.1 Háttér

A JServ-et helyettesítették az Apache Jakarta projektben Tomcat-el: <http://jakarta.apache.org/> . Ez a rész csak a JServvel foglalkozik és frissítésre szorul.

Az Apache JServ egy 100%-osan tiszta Java szervlet motor, amely teljesen együttműködik a Java Servlet 2.0 specifikációval. Az Apache JServ a Java Apache Project része. (Az Apache Java Project weboldaláról.)

4.2.2 Letöltés

Apache JServ letölthető a <http://java.apache.org/jserv/index.html> oldalról.

Az Apache JServ Project weboldaláról indulva kövesd a Download Apache JServ linket.

Jelenleg RPM csomagok állnak rendelkezésre RedHat Linuxra. Más Linux disztribúciókhoz neked kell beépítened a forrást. A következő példa megmutatja, hogy hogyan telepítsd az RPM-et RedHat 6x verzióra.

A doksi írásakor a legfrissebb verzió az 1.1b3-as.

4.2.3 Telepítés

Az javasolom, hogy telepítsd a fájlokat az /usr/local könyvtárba. Miután letöltötted őket, futtasd a következő parancsot:

```
mkdir /usr/local/apachejserv
mv ApacheJServ*.rpm /usr/local/apachejserv
```

RedHat és RedHat alapú disztribúciók esetén használd az RPM-et a csomag telepítéséhez, valahogy így:

```
rpm -i ApacheJServ-1.1-b2_RH6x.i386.rpm
```

A fenti példák az 1.1-b2 verzióra készültek RedHat 6x alá, Intel architektúrára.

4.2.4 Környezeti beállítások

Le kell állítanod, hogy beállíthasd a környezeti változókat, majd újra kell indítanod az Apache kiszolgálót az Apache Jserv-ed regisztrálásához.

Az Apache leállításához, írd be:

```
cd /etc/rc.d/init.d
./httpd stop
```

Megjegyzés: A httpd szkript bootoláskor lehet, hogy máshol található a különböző Linux disztribúciónál.

A Java környezet beállításához nézd meg a Hogyan telepítsünk JDK-t részt ebben a leírásban, az általad használt JDK-ra való tekintettel. Be kell állítanod néhány tulajdonságot a jserv fájlban, ami az /etc/httpd/conf/jserv könyvtárban található.

Különös tekintettel az alábbiakra:

- wrapper.bin - A JDK helyére hivatkozva

- wrapper.classpath - minimálisan /usr/lib/apache/ApacheJServ.jar and /home/httpd/classes/servlet-2.0.jar
- bindaddress=localhost
- port=8007

Az Apache újraindításához, írd be:

```
cd /etc/rc.d/init.d
./httpd start
```

4.2.5 Telepítés ellenőrzése

Az Apache JServ telepítésének és működésének ellenőrzéséhez nyisd meg a böngésződ és írd be az URL-be: "http://127.0.0.1/servlet/IsItWorking" . Helyettesítsd a megfelelő IP címet, ha egy másik gépről böngészel.

Egy "Yes, Its Working!" weboldalt kell látnod.

Gratulálok, telepítetted, beállítottad, tesztelted az Apache JServ-et Linux alatt! További segítségért látogass el az Apache JServ weboldalára a <http://java.apache.org/jserv/index.html> címre.

Most fordítsd és futtasd a saját kis szervleted. Írd be a következő Java szervlet programot:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorldServlet extends HttpServlet {
    public void service (HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
    }
}
```

4.2.6 További információk

További információért látogass el a Java Apache Project weboldalára a <http://java.apache.org/> címre.

4.3 BEA WebLogic

Lásd [7.1](#) (BEA WegLogic) lejjebb.

4.4 Enhydra

A leírás elkészítése folyamatban van.

További információ található a <http://www.enhydra.org> oldalon.

4.5 IBM WebSphere

A leírás elkészítése folyamatban van.

További információ található a <http://www-4.ibm.com/software/webservers/appserv/linux.html> oldalon.

4.6 Locomotive

A leírás elkészítése folyamatban van.

További információ található a <http://www.locomotive.org/> oldalon.

4.7 Jetty

A Jetty http szerver egy kombinált szerver és szervlet konténer. A http szerver telepítése (lásd fentebb) szervlet támogatást is biztosít. További információ elérhető a demó szerveren és az oktatón keresztül, amit a http szerverrel együtt telepítettél.

5 Hogyan telepíts Java Server Pages (JSP) támogatást

A leírás elkészítése folyamatban van.

5.1 Apache Jakarta

A leírás elkészítése folyamatban van.

További információ található a <http://jakarta.apache.com/> oldalon.

5.2 Caucho Resin

A leírás elkészítése folyamatban van.

További információ található a <http://www.caucho.com/> oldalon.

5.3 Jetty

A Jetty HTTP szerver a Jasper JSP motorral rendelkezik. A HTTP szerver telepítése, lehetőséget ad (lásd fent 3.4) JSP támogatásra. További információ elérhető a demó szerveren és az oktatón keresztül, amit a http kiszolgálóval együtt telepítettél.

6 JDBC támogatás telepítése

Van néhány adatbázis, ami fut Linux alatt és támogatja a JDBC interfészt. Ezek például:

- [6.1](#) (IBM DB2)
- [6.2](#) (MiniSQL)
- [6.3](#) (MySQL)
- [6.4](#) (Oracle)

- 6.5 (PostgreSQL)
- 6.6 (Sybase)

Ha csak egy DBMS-t szeretnél kipróbálni, akkor először a PostgreSQL-el próbálkozz, ez minden nagyobb Linux-disztribúcióban benne van és lehet, hogy már telepítve van a rendszerre.

6.1 IBM DB2

A leírás elkészítése folyamatban van.

További információ található a <http://www-4.ibm.com/software/data/db2/linux/> oldalon.

6.2 MiniSQL

A leírás elkészítése folyamatban van.

További információ található a <http://www.hughes.com.au/> oldalon.

6.3 MySQL

A leírás elkészítése folyamatban van.

További információ található a <http://www.mysql.org/> oldalon.

6.4 Oracle

A leírás elkészítése folyamatban van.

További információ található a <http://platforms.oracle.com/linux/> oldalon.

6.5 PostgreSQL

6.5.1 Háttér

A PostgreSQL egy kifinomult objektum-relációs alapú DBMS, amely támogatja a legtöbb SQL konstruktort, subselect-et, tranzakciókat és a felhasználó által definiált típusokat és funkciókat. Ez a jelenlegi legjobb nyílt forráskódú adatbáziskezelő, ami elérhető. Kereskedelmi forgalomban is kapható a PostgreSQL Inc-től. A jelenlegi verzió a 6.5.3-as és elérhető bármelyik tüköroldalról vagy CD-ről. (A PostgreSQL weboldaláról)

Lehetséges, hogy a PostgreSQL már a Linux disztribúcióban benne van, hiszen ez is nyílt forráskódú.

6.5.2 Letöltés és telepítés

Azt javasolom, hogy kezdetben inkább azt a PostgreSQL-t használd, amelyik a Linux disztribúcióban van, mintsem egy letöltött verziót.

Vagy ha mégsem azt akard használni, akkor megtalálod a PostgreSQL-t a <http://www.postgresql.org> oldalon.

Annak ellenőrzéséhez, hogy a PostgreSQL már telepítve van-e a gépedre, írd be a következő parancsot:

```
rpm -qa | grep postgresql
```

vagy

```
which postmaster
```

```
which psql
```

Szükséged lesz a telepített postgresql, postgresql-szerver, és postgresql-java csomagokra, hogy a Javát együtt használd a PostgreSQL-el.

Ellenőrizd, hogy fut-e a PostgreSQL:

```
ps -f -u postgres
```

Most látnod kell a postmaster, a PostgreSQL démon futását.

Ha a postmaster nem fut, akkor van egy Sys V Init szkript, amivel elindíthatod. Ez a legtöbb disztribúcióban az /etc/rc.d/init.d könyvtárban található. A PostgreSQL indításához írd be:

```
cd /etc/rc.d/init.d  
./postgresql start
```

Használhatod a fentebb már leírt ps parancsot, a PostgreSQL futásának ellenőrzéséhez.

Megjegyzés: A JDBC használatához a PostgreSQL-t -i -vel kell indítanod! Egyébként csak UNIX domain socket-eket fog támogatni, TCP/IP kapcsolatot nem. Ellenőrizd, hogy a postmaster az -i paraméterrel indult!

Készíts egy próba adatbázist:

```
su - postgres  
createdb javatest
```

Nem szabad hibaüzenetet kapnod.

Készíts egy próba táblát egy próba sorral. Elsőként jelentkezz be az interaktív PostgreSQL felületre és kapcsolódj a javatest adatbázishoz, amit készítettél:

```
psql javatest
```

Egy megerősítést kell látnod, hogy kapcsolódtál a javatest adatbázishoz. Azután készíts egy próba táblát (psql nélkül):

```
create table test (col1 varchar(255));
```

A "CREATE" üzenetet kell látnod, ha elkészült.

A következő lépésben helyezz el benne egy sort (psql nélkül):

```
insert into test (col1) values ('Hello, from PostgreSQL!');
```

Egy "INSERT" válaszüzenetet kell kapnod.

Végül ellenőrizd, hogy a sor a helyén van (psql nélkül):

```
select col1 from test;
```

Most látnod kell a lekérdezés eredményét. A kilépéshez írd be, hogy "\q".

További segítségért nézd meg a Database-SQL-RDBMS HOW-TO doksit a <http://metalab.unc.edu/mdw/HOWTO/PostgreSQL-HOWTO.html>

címen.

A szükséges JAR-okat hozzá kell adnod a CLASSPATH változóhoz. A PostgreSQL JAR-ok a *postgresql-jdbc* csomagban vannak.

```
export CLASSPATH=$CLASSPATH:/usr/lib/pgsql/jdbc6.5-1.2.jar
```

Helyettesítsd a megfelelő útvonalat arra ahol a te PostgreSQL csomagod van.

6.5.3 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egy egyszerű JDBC alkalmazást, ami PostgreSQL-t használ. Készítsd el a következő programot:

```
import java.sql.*;

class PostgreSQLTest {
    public static void main (String[] args) {
        try {
            Driver driver = (Driver)
                Class.forName("postgresql.Driver").newInstance();
            DriverManager.registerDriver(driver);

            String url = "jdbc:postgresql:javatest";
            Connection con = DriverManager.getConnection(url, "postgres", "");
            Statement stm = con.createStatement();

            stm.setQueryTimeout(10);
            ResultSet rs = stm.executeQuery("select col1 from test");

            rs.next();

            System.out.println(rs.getString(1));

        } catch (SQLException e) {

            System.out.println("Exception!");
            System.out.println(e.toString());
        }
    }
}
```

Fordítsd le a Java fordítóval.


```
javac PostgreSQLTest.java
```

Ha a fordító hibát produkál, ellenőrizd duplán a szintaxist és ellenőrizd a PATH és a CLASSPATH változók beállításait.

Futtasd a programot a JVM-el.

```
java PostgreSQLTest
```

Ha a JVM hibát produkál, akkor ellenőrizd a PATH és a CLASSPATH útvonalakat.

A következőt kell látnod:

```
Hello, from PostgreSQL!
```

Gratulálok, telepítetted, beállítottad és tesztelted a JDBC interfészt a PostgreSQL-el!

6.5.4 További információk

További információért keresd fel a PostgreSQL weboldalát a <http://www.postgresql.org/> . oldalon.

6.6 Sybase

6.6.1 Háttér

A Sybase Adaptive Server Enterprise egy kereskedelmi forgalomban lévő RDBMS, ami elérhető Linux operációs rendszerre. A Sybase jelenleg kiadott verziója a 12.0-ás, Linuxra még csak a 11.9.2 érhető el.

Idézet a Sybase weboldaláról: "Az ASE-t a Linuxhoz implementálva, a Sybase az első magasan skálázható, nagy hatékonyságú adatbázis motort adja a Linux fejlesztők kezébe. A csomag tartalmazza az Adaptive Server Enterprise irányadó sajátosságait és a kapcsolódó komponenseket. Az Adaptive Server Enterprise fejlesztésre szabadon használható."

6.6.2 Letöltés

A Sybase ASE letölthető a http://www.sybase.com/products/databaseservers/linux/linux1192_reg.html oldalról.

A letölthöz regisztrálnod kell magad a Sybase-nél és el kell fogadnod a licenc feltételeit.

A Sybase JDBC meghajtó letölthető a <http://www.sybase.com/products/internet/jconnect/> oldalról.

A letöltésnél válaszd a jConnect 4.2/5.2 csomagot.

Ha van hozzáférése a Sybase szerver hálózatához, akkor csak letöltened és telepítened kell a JDBC meghajtót.

6.6.3 Telepítés

A Sybase telepítése ezen HOGYAN hatókörén kívül esik. Ez a HOGYAN feltételezi, hogy a Sybase-t rendben telepítetted és konfiguráltad, és kész vagy a Sybase isql használatára.

Jelentkezz be az isql-be mint rendszer adminisztrátor és készíts egy teszt felhasználót és adatbázist.

```
create database javatest
go
sp_addlogin javatest, javatest, javatest
go
use javatest
go
sp_downer javatest
go
```

Nem szabad hibaüzenetet kapnod.

Készíts egy próba táblát egy próba sorral. Először jelentkezz be az isql-be mint a javatest felhasználója és írd be a következőt:

```
create table test (col1 varchar(255))
go
```

Nem szabad hibaüzenetet kapnod.

Vegyél fel egy rekordot:

```
insert into test (col1) values ('Hello, from Sybase!')
go
```

Nem szabad hibaüzenetet kapnod.

Végül ellenőrizd, hogy a rekord a helyén van:

```
select col1 from test
go
```

Most azt a sort kell látnod, amit bevittél.

Az isql-ből az "exit" beírásával tudsz kilépni.

További segítségért a Sybase-el való munkához nézd át a dokumentációját, ami letölthető.

Szükséged lesz a megfelelő JAR-okra a CLASSPATH változóban.

```
export CLASSPATH=$CLASSPATH:/usr/local/sybase/jConnect-5_2/classes/jconn2.jar
```

A megfelelő útvonalat helyettesítsd, attól függően, hogy a jConnect hová települt a rendszeredben.

6.6.4 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egy egyszerű JDBC alkalmazást, ami Sybase-t használ. Készítsd el a következő programot.

```
import java.sql.*;

class SybaseTest {
    public static void main (String[] args) {
        try {
```

```
Driver driver = (Driver)
    Class.forName("com.sybase.jdbc2.jdbc.SybDriver").newInstance();
DriverManager.registerDriver(driver);

String    host = "127.0.0.1";
String    port = "4100";

String    url = "jdbc:sybase:Tds:" + host + ":" + port;
Connection con = DriverManager.getConnection(url, "javatest", "javatest");
Statement stm = con.createStatement();

stm.setQueryTimeout(10);
ResultSet rs = stm.executeQuery("select col1 from test");

rs.next();

System.out.println(rs.getString(1));

} catch (SQLException e) {

    System.out.println("Exception!");
    System.out.println(e.toString());
}
}
```

Ahol szükséges helyettesítsd a host-ot és a port számot. Nézd meg a \$SYBASE/interfaces és a \$DSQUERY belépéseit a host és port beállításához.

Fordítsd le a programot a Java fordítóval.

```
javac SybaseTest.java
```

Ha a fordító hibát produkál, akkor nézd meg a szintaxist és ellenőrizd a PATH és a CLASSPATH változók beállításait.

Futtasd a programot a JVM-mel.

```
java SybaseTest
```

Ha a JVM hibát produkál, akkor ellenőrizd a PATH és a CLASSPATH változók beállításait.

A következő eredményt kell látnod:

```
Hello, from Sybase!
```

Gratulálok, telepítetted, beállítottad és tesztelted a JDBC interfészt a Sybase-hez.

6.6.5 További információk

További információért látogass el a Sybase jConnect weboldalára a <http://www.sybase.com/products/internet/jconnect/> helyen.

7 Hogyan telepítsük az Enterprise Java Bean (EJB) támogatást

A leírás elkészítése folyamatban van.

7.1 BEA WebLogic

A leírás elkészítése folyamatban van.

További információ található a <http://www.beasys.com/linux/> honlapon.

7.2 EJBoss

7.2.1 Háttér

Az EJBoss a JBoss-tól kapta a nevét és nagyon jól működik együtt a stabil J2EE verziókkal: <http://www.jboss.org/>.

Ez a rész akkor készült, amikor még csak az EJBoss 0.95-ös verziója jelent meg, ezért feltétlenül frissítsd az információkat.

7.2.2 Letöltés

A JBoss letölthető a JBoss weboldaláról: <http://www.jboss.org/>.

7.2.3 Telepítés

Azt javaslom, hogy telepítsd a fájlokat az `/usr/local` könyvtárba. A letöltés után futtasd a következő parancsokat:

```
mkdir /usr/local/ejboss
mv ejboss* /usr/local/ejboss
```

Csomagold ki a fájlt:

```
jar xvf ejboss095_jdk122.jar
```

Mindenféle fájlt és könyvtárat kell látnod az `/usr/local/ejboss` könyvtárban.

A fenti példa az EJBoss 0.95-ös verziója alapján készült. Helyettesítsd a fájlneveket, ahol szükséges.

7.2.4 Környezeti beállítások

A következő környezeti változókat kell beállítanod:

- CLASSPATH

A CLASSPATH változó mutatja meg, hogy mely JAR-okra és könyvtárakra lesz szükséged, hogy fordíts és futtass Java programokat. A CLASSPATH feltétlenül tartalmazza az EJBoss JAR-t és a `/bean/generated` könyvtárakat.

```
export CLASSPATH=/usr/local/ejboss/lib/ejboss095_jdk122.jar:/usr/local/ejboss/beans/generated:$CLASS
```

7.2.5 A telepítés ellenőrzése

Most már fordíthatsz és futtathatsz egy egyszerű EJB alkalmazást. Készítsd el a következő három forrásfájlt a szerver számára.

Először a business interfészt.

```
// EJBTest.java

import javax.ejb.*;
import java.rmi.RemoteException;

public
    interface EJBTest extends EJBObject {
        public String greet() throws
            RemoteException;
    }
```

Másodszor, a home interfészt.

```
// EJBTestHome.java

import javax.ejb.*;
import java.rmi.RemoteException;

public
    interface EJBTestHome extends EJBHome {

        public EJBTest create() throws

            CreateException, RemoteException;
    }
```

Harmadszor, a bean implementációs osztályt.

```
// EJBTestBean.java

import javax.ejb.*;
import java.rmi.RemoteException;

public
    interface EJBTestBean implements SessionBean {

        private SessionContext
            mContext = null;

        public void ejbPassivate() {
            System.out.println("EJBTestBean
                passivated.");
        }
    }
```

```
public void ejbActivate() {
    System.out.println("EJBTestBean
activated.");
}

public void ejbCreate() {
    System.out.println("EJBTestBean
created.");
}

public void ejbRemove() {
    System.out.println("EJBTestBean
removed.");
}

public void setSessionContext() {
    System.out.println("EJBTestBean
context set.");
    mContext = context;
}

public String greet()
{
    return "Hello, I'm an EJB!";
}
}
```

Fordítsd le a szerver forrásfájljait a Java fordítóval:

```
javac EJBTest*.java
```

Ha a fordító hibát produkál, akkor nézd meg kétszer is a szintaxist és ellenőrizd a PATH és a CLASSPATH változók beállításait.

Most, hogy megírtad és lefordítottad a szerver forrásfájljait, töltsd fel a beaneket az EJBoss-ba. Ez néhány nagyon fontos lépésből áll, amelyekre figyelni kell.

Először készítsd el az ejb-jar.xml fájlt.

```
<?xml version="1.0" encoding="Cp1252"?>
```

```
<ejb-jar ID="">
```

```
    <description></description>
```

```
    <display-name></display-name>
```

```
    <small-icon></small-icon>
```

```
<large-icon></large-icon>

<ejb-client-jar></ejb-client-jar>

<enterprise-beans>

  <session>

    <description>Nextgen bean</description>

    <ejb-name>nextgen.EJBTest</ejb-name>

    <home>EJBTestHome</home>

    <remote>EJBTest</remote>

    <ejb-class>EJBTestBean</ejb-class>

    <session-type>Stateful</session-type>

    <transaction-type>Bean</transaction-type>

    <env-entry>

      <description></description>

      <env-entry-name></env-entry-name>

      <env-entry-type>java.lang.String</env-entry-type>

      <env-entry-value></env-entry-value>

    </env-entry>

    <resource-ref>

      <description></description>

      <res-ref-name></res-ref-name>

      <res-type></res-type>

      <res-auth>Container</res-auth>

    </resource-ref>

  </session>

</enterprise-beans>
```

```
<assembly-descriptor />

</ejb-jar>
```

A fenti ejb-jar.xml fájl azonosítja az interfészt és az osztályneveket, amelyeket készítettél. Másodszor, a három "class" fájl könyvtárában készíts egy META-INF könyvtárat.

```
mkdir META-INF
mv ejb-jar.xml META-INF
```

Harmadszor, csomagold a fájlokat egy jar-ba.

```
jar cvf EJBTest.jar EJBTest*.class META-INF/ejb-jar.xml
```

Látnod kellene, hogy hozzáadódott a listához, csakúgy mint a három "class" fájl és az XML betöltést leíró fájlt.

Negyedszer, tedd a most készült Jar-t az EJBoss bean könyvtárába.

```
mv EJBTest.jar /usr/local/ejboss/beans
```

Ötödször, másold az osztályfájlokat, amiket készítettél, az EJBoss beans/generated könyvtárba.

```
mv EJBTest*.class /usr/local/ejboss/beans/generated
```

(Az ötödik lépés felesleges egy hiba miatt, ami az EJBoss 0.95-ben van.)

Most már elindíthatod az EJBoss kiszolgálót.

```
cd /usr/local/ejboss
sh server.sh
```

Azt kell látnod, hogy a proxy fájlok automatikusan lefordítódnak és megerősítik, hogy az EJB-d betöltődött.

Most már írhat, fordíthat és tesztelhet egy egyszerű kliens alkalmazást.

7.3 Bullsoft JOnAS EJB

A leírás elkészítése folyamatban van.

További információ található a <http://www.bullsoft.com/ejb/> oldalon.