

Programming Bileşeni Test Aşamaları

Semen Cirit

20 Ağustos 2009

1 Tool alt bileşeni

1. fcgi sadece kurulum testine tabidir.

2 vcs alt bileşeni

1. mod_dav_svn paketi sadece kurulum testine tabidir.

2. git paketi kurulumu sonrası:

Aşağıdaki komutları çalıştırın. Ve sorunsuz bir şekilde Git deposu oluşturduğunu ve klonlandığını gözlemleyin.

```
# cd ~
# mkdir test_git
# cd test_git
# git init
# cd ..
# git clone test_git test_clone
```

3. subversion paketi kurulumu sonrası:

test dizininin sorunsuz bir şekilde eklenmiş olduğunu gözlemleyin:

```
# svn co http://svn.pardus.org.tr/uludag/trunk/test/2009/testguide/turkish/
# cd turkish
# svn mkdir test
# svn st
```

testfile dosyası içerisine bir kaç kelime yazın ve kaydedin. Yapılan değişiklik farkının alınabildiğini gözlemleyin:

```
# vi testfile
# svn add testfile
# svn diff
```

3 Environment alt bileşeni

1. eric paketi kurulumu sonrası:

Aşağıda bulunan dosyayı eric uygulaması ile açın ve Start → Run Script yolunu izleyerek çalıştırın.

Sorunsuz bir şekilde çalıştığını gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/environment/test.py
```

2. Aşağıda bulunan paketlerin kurulumu sonrasında, yerel dilinizi değiştirip, konsoldan aynı dizinde bir open office uygulaması açın ve yardım dosyasının ilgili dilde olduğunu gözlemleyin.

```
eric-i18n-cs
eric-i18n-de
eric-i18n-es
eric-i18n-fr
eric-i18n-ru
eric-i18n-tr
```

Yerel dili değiştirmek için:

```
export LC_ALL= <lang_LANG>
```

lang_LANG şeklinde yazılmış olan, pt-BT için pt_BT, diğer diller için örneğin de_DE olacaktır.

Daha sonra bu çalıştırdığımız komut dizininde eric4 komutunu çalıştırın, paket eğer help ile ilgili ise help dosyasının, uygulama dili ise uygulamanın sorunsuz bir şekilde istenilen dilde açıldığını gözlemleyin.

3. ipython paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırdığınızda, bulunduğunuz dizinde test adında bir dosya oluştuğunu ve içerisinde "test ipython" yazdığını gözlemleyin:

```
# ipython
a = open("test", "a")
a.write("test ipyton")
```

4. drscheme paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde çalıştığını gözlemleyin.

5. qt-creator paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde açıldığını gözlemleyin.

4 Language alt bileşeni

4.1 Perl alt bileşeni

1. perl-IO-Socket-SSL paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/IO-Socket-SSL-1.26.tar.gz
```

Konsoldan;

```
# cd IO-Socket-SSL-1.26/
# /usr/bin/perl5.10.0 "-MExtUtils::Command::MM" "-e" "test_harness(0,'blib/lib', 'blib/arch')" t/*.t
```

Komutlarını çalıştırın ve testlerden "ok" sonuçlarının döndüğünü gözlemleyin.

2. perl-Compress-Zlib paketi kurulumu sonrası:

programming-tr.pdf git testini gerçekleştirin.

3. perl-Email-MIME-Encodings paketi kurulumu sonrası:

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/Email-MIME-Encodings.t
# perl Email-MIME-Encodings.t
```

Tüm sonuçların "ok" döndürdüğünü gözlemleyin.

4. perl-Email-MIME-Encodings paketi kurulumu sonrası:

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/test_perl_Test_Simple.t
# perl test_perl_Test_Simple.t
```

Tüm sonuçların "ok" döndürdüğünü gözlemleyin.

4.2 Python alt bileşeni

1. Django paketi kurulumu sonrası:

- Aşağıda bulunan komutu çalıştırın:

```
# django-admin.py startproject test
# cd test
```

test adında bir dizinin oluştuğunu ve bu dizin altında aşağıda bulunan dosyaların oluştuğunu gözlemleyin.

```
__init__.py
manage.py
settings.py
urls.py
```

- Aşağıdaki komutu çalıştırın ve daha sonra firefox'tan `http://localhost:8080/` adresini girin ve sunucuya bağlanabildiğinizi gözlemleyin

```
# python manage.py runserver 8080
```

- settings.py içerisinde DATABASE_ENGINE DATABASE_NAME değişkenlerini aşağıdaki database değişkenlerini atayın:

```
DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = 'sqlite3_'
```

- Aşağıdaki komutu çalıştırın ve istemiş olduğu işlemleri sırasıyla gerçekleştirin ve sorunsuz bir şekilde Django onay sisteminin kurulduğunu gözlemleyin:

```
# python manage.py syncdb
```

- Aşağıdaki komutu çalıştırın ve polls adında bir dizinin oluştuğunu gözlemleyin:

```
# python manage.py startapp polls
```

- Ve dizin içerişinde aşağıdaki dosyaların oluştuğunu gözlemleyin:

```
__init__.py
models.py
views.py
```

2. python-memcached paketi kurulumu sonrası:

Apache sunucusunu servis yöneticisinden başlatın.

Aşağıdaki komutları çalıştırın ve sonucun "True" döndürdüğünü gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/
python/test_python-memcache.py
# python test_python-memcache.py
```

3. pygtk paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırın ve sorunsuz bir şekilde bir pencerenin açıldığını gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/desktop/toolkit/test_pango.py
# python test_pango.py
```

4. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import mpmath
```

5. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import mpmath
```

6. python-M2Crypto paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import M2Crypto
```

7. winpdb paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import winpdb
```

(DeprecationWarning önemli değildir.)

8. cython paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import cython
```

9. lxml paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import lxml
```

10. python-RuleDispatch paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import dispatch
```

11. python-nose paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import nose
```

12. PyICU paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import PyICU
```

13. python-simplejson paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import simplejson
```

4.3 Java alt bileşeni

1. Aşağıdaki paketlerin kurulumu sonrası:

```
sun-jre
sun-jdk
sun-jdk-demo
sun-jdk-samples
sun-jdk-doc
```

Aşağıda bulunan komutların düzgün bir şekilde çalıştığını gözlemleyin.

```
# java -version
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/java/test.java
# javac test.java
# java test
```

4.4 Lisp alt bileşeni

1. clisp paketi kurulumu sonrası: (Warningleri önemsemeyiniz.)

Aşağıdaki komutların çalıştırın ve hata olmadığını gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/lisp/test_clisp.lisp
# clisp -c test_clisp.lisp
```

4.5 Dotnet alt bileşeni

1. mono paketi kurulumu sonrası: Aşağıdaki komutların çalıştırın ve hata olmadığını gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/dotnet/test_mono.cs
# mcs test_mono.cs
# mopno test_mono.exe
```

- R paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırın ve bir grafiğin oluştuğunu gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/test_R.R
# R --vanilla --slave < test_R.R
```

- R-mathlib paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırın ve sorunsuz bir şekilde çalıştıklarını gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/test_r-mathlib.c
# gcc -o test_r-matlib test_r-matlib.c -lm -lRmath
```