

Programming Bileşeni Test Aşamaları

Semen Cirit

18 Eylül 2009

1 Microcontroller alt bileşeni

1. Aşağıda bulunan paketler kurulum testine tabidir.

```
avr-libc
avrdude
binutils-avr
gcc-avr
```

2 Tool alt bileşeni

1. fci sadece kurulum testine tabidir.

3 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
```

4 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
```

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

- ipython-gui paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırdığınızda ipython guilerinin sorunsuz bir şekilde açıldığını gözlemleyin.

```
# ipython-wx
# ipythonx
```

- 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 ipython")
```

- drscheme paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde çalıştığını gözlemleyin.

- qt-creator paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde açıldığını gözlemleyin.

5 Language alt bileşeni

5.1 Php alt bileşeni

- php-cli ve php-common paketleri kurulumu sonrası:

Aşağıda bulunan komutları çalıştırdıktan sonra <http://localhost/test.php> adresine firefox ile girin ve php ile ilgili bilgilerin sayfalandığını gözlemleyin.

```
# cd /var/www/localhost/htdocs/
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/php/test.php
```

5.2 Perl alt bileşeni

- 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.

5.3 Python alt bileşeni

1. pyFltk paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import fltk
```

2. python-turboflot paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import turboflot
```

3. python-ldap paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import ldap
```

4. python-iptables paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import iptables
```

5. sympy paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırdığınızda testlerden ok sonucu döndüğünü gözlemleyin.

```
# python -c "import sympy;print sympy.test()"
```

```
python -c "import scipy;print scipy.test()"
```

6. scipy paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırdığınızda `jnose.result.TextTestResult run=XXXX errors=0 failures=0` gibi bir çıktı ürettiğini gözlemleyin.

```
# python -c "import scipy;print scipy.test()"
```

7. PyX paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pyx
```

8. pyNotifier paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pynotify
```

9. httpLib2 paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import httpLib2
```

10. 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 içeriğinde aşağıdaki dosyaların oluştuğunu gözlemleyin:

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

11. 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
```

12. 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
```

13. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import mpmath
```

14. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import mpmath
```

15. python-M2Crypto paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import M2Crypto
```

16. winpdb paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import winpdb
```

(DeprecationWarning önemli değildir.)

17. cython paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import cython
```

18. lxml paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import lxml
```

19. python-RuleDispatch paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import dispatch
```

20. python-nose paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import nose
```

21. PyICU paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import PyICU
```

22. python-simplejson paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import simplejson
```

5.4 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
```

5.5 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
```

5.6 Dotnet alt bileşeni

1. Aşağıda bulunan paketler sadece kurulum testine tabidir

```
taglib-sharp
```

2. gmime, gmime-docs, gmime-sharp paketi kurulumu sonrası:

Aşağıda bulunan komutun jpeg dosyasını encode ettiğini gözlemleyin.

```
# wget http://cekirdek.pardus.org.tr/~semen/dist/test/multimedia/graphics/test_dcraw.jpg
# gmime-uuencode -m test_dcraw.jpg jpeg
```

3. 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
```