Programming Companent Testcases

Semen Cirit

30 Eylül 2009

1 Microcontroller alt bileşeni

1. Following packages subject to installation test:

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

2 Tool sub component

1. Following packages subject to installation test:

```
fcgi
```

3 vcs sub component

1. mod_dav_svn package depended installation test.
2. After installation git package:
   Run below commands. And observe that it creates Git deposu and clones it correctly.

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

3. After installation subverison package:
   Observe that test directory added correctly:

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

   Write some words to testfile and save it. observe that you fetch change difference.

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

4 Environment sub component

1. After installation eric package:

   Open the below file with eric application and run it following Start → Run Script way.

   Observe that it runs correctly.

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

2. After installation below packages, change local language, open an open office application from console at same directory and observe that help file's language be same depended language.

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

   for changing local language:

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

   represented by lang_LANG, it will be pt-BT for pt_BT, for other languages example de_DE

   After run eric4 command at the command directory, 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.

3. After installation ipython package:

   Observe that when you run below commands, 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. After installation drscheme package:

   Open the application from Kmenu and and observe that it runs correctly.

5. After installation qt-creator package:

   Open the application from Kmenu and and observe that it runs correctly

5 Language sub component

5.1 Php sub component

1. After installation php-cli and php-common packages:

   After run below commands, enter http://localhost/test.php adress by firefox and observe that it paged informations about php.

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

5.2 Perl sub component

1. After installation perl-IO-Socket-SSL package: Download below file and open it.

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

   from console;

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

run the commands and observe that returned "ok" from tests.

2. perl-Compress-Zlib paketi kurulumu sonrası:

   Do programming-eng.pdf git test.

3. After installation perl-Email-MIME-Encodings package:

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

   observe that all results returned "ok".

4. After installation perl-Email-MIME-Encodings package:

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

   observe that all results returned "ok".

5.3 Python sub component

1. After installation pyFltk package:

   İnstall ipython package and run the following commands:

```
# ipython
import fltk
```

2. After installation python-turboflot package:

   İnstall ipython package and run the following commands:

```
# ipython
import turboflot
```

3. After installation python-ldap package:

   İnstall ipython package and run the following commands:

```
# ipython
import ldap
```

4. After installation python-iptables package:

   İnstall ipython package and run the following commands:

```
# ipython
import iptables
```

5. After installation sympy package:

   Observe that the results return ok when you run the below command.

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

6. After installation scipy package:

   Observe that the results the below command returns a result like "¡nose.result.TextTestResult run=XXXX errors=0 failures=0¿".

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

7. After installation PyX package:

   İnstall ipython package and run the following commands:

```
# ipython
import pyx
```

8. After installation pyNotifier package:

   İnstall ipython package and run the following commands:

```
# ipython
import pynotify
```

9. After installation httplib2 package:

   İnstall ipython package and run the following commands:

```
# ipython
import httplib2
```

10. After installation Django package:

    - run below commands:

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

      observe that it creates test named directory and creates below files under that directory.

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

    - run below commands and then enter http://localhost:8080/ adress from firefox and observe that you can connect to server.

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

    - assign below database variables to DATABASE_ENGINE DATABASE_NAME variables in settings.py:

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

    - run below command ve istemiş olduğu işlemleri sırasıyla gerçekleştirin ve sorunsuz bir şekilde Django onay sisteminin kurulduğunu gözlemleyin:

      `# python manage.py syncdb`

    - run below command and observe that it creates poll named directory:

      `# python manage.py startapp polls`

    - and observe that it creetes below files in the directory:

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

11. After installation python-memcached package:

    start Apache server from service manager.

    run below commands and observe that it's result returned "true".

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

12. After installation pygtk package:

    run below commands and observe that opened a window correctly.

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

13. After installation mpmath package:
    install ipython package and run below commands:

    ```
    # ipython
    import mpmath
    ```

14. After installation python-M2Crypto package:
    install ipython package and run below commands:

    ```
    # ipython
    import M2Crypto
    ```

15. After installation winpdb package:
    install ipython package and run below commands:

    ```
    # ipython
    import winpdb
    ```

    (DeprecationWarning isn't important.)

16. After installation cython package:
    install ipython package and run below commands:

    ```
    # ipython
    import cython
    ```

17. After installation lxml package:
    install ipython package and run below commands:

    ```
    # ipython
    import lxm
    ```

18. After installation python-RuleDispatch package:
    install ipython package and run below commands:

    ```
    # ipython
    import dispatch
    ```

19. After installation python-nose package:
    install ipython package and run below commands:

    ```
    # ipython
    import nose
    ```

20. After installation PyICU package:
    install ipython package and run below commands:

    ```
    # ipython
    import PyICU
    ```

21. After installation python-simplejson package:
    install ipython package and run below commands:

    ```
    # ipython
    import simplejson
    ```

5.4 Java sub component

1. After below packages installation:

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

   observe that they run below commands correctly.

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

5.5 Lisp sub component

1. After installation clisp package: (don't worry about warnings.)
   run below commands and observe that no error.

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

5.6 Dotnet sub component

1. Following packages subject to installation test:

   ```
   taglib-sharp
   ```

2. After installation gmime, gmime-docs, gmime-sharp packages:
   Observe that the following command encode the jpeg file.

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

3. After installation mono package:
   run below commands and observe that no error.

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

- After installation R package:
  run below commands and observe that it creates a graph.

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

- After installation R-mathlib package:
  run below commands and observe that they run correctly.

  ```
  # 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
  ```