

# Programming Bileşeni Test Aşamaları

Semen Cirit

7 Ocak 2010

## 1 Profiler alt bileşeni

1. valgrind paketi kurulumu sonrası:

Gimp uygulamasının işlem ve hafıza durumunun listelendiğini gözlemleyin.

```
valgrind gimp
```

2. oprofile paketi kurulumu sonrası:

Aşağıda bulunan komutların hatasız çalıştığını gözlemleyin.

```
su -
opcontrol --no-vmlinux
opcontrol --start
opcontrol --dump
opreport
```

## 2 Debug alt bileşeni

1. gdb paketi kurulumu sonrası:

Aşağıda bulunan komutları çalıştırın ve gdb-test.c'in derlendiğini ve daha sonra gdb-test ile ilgili gdb'nin bir hata bulduğunu gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/debug/gdb-test.c
gcc -g gdb-test.c -o gdb-test
./gdb-test
gdb gdb-example
run
backtrace
quit
```

## 3 Microcontroller alt bileşeni

1. Aşağıda bulunan paketler kurulum testine tabidir.

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

## 4 Tool alt bileşeni

1. Aşağıda bulunan paketler sadece kurulum testine tabidir.

```
fcgi
mcpp
```

2. translate-toolkit paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import translate
```

3. cdecl paketi kurulumu sonrası:

Aşağıda bulunan komutların düzgün çalıştığını gözlemleyin.

```
# cdecl
cdecl> #include <string.h>
```

## 5 vcs alt bileşeni

1. Aşağıda bulunan paketler sadece kurulum testine tabidir.

```
mod_dav_svn
abichack
git-cvs
git-emacs
git-gui
git-svn
gitweb
```

2. SVK paketi kurulumu sonrası:

Aşağıda bulunan komutların sorunsuz çalıştığını gözlemleyin:

```
svk mirror //svk/trunk svn://svn.clkao.org/svkbook/trunk
svk sync //svk/trunk
```

3. tig paketi kurulumu sonrası:

tig komutunun tüm geçmişi listelediğini gözlemleyin.

```
git clone git://github.com/git/hello-world.git
cd hello-world
tig
```

4. mercurial paketi kurulumu sonrası:

Aşağıda bulunan komutların sorunsuz çalıştığını gözlemleyiniz.

```
hg clone http://selenic.com/hg mercurial-repo
cd mercurial-repo
hg parents
```

5. gitk paketi kurulumu sonrası:

gitk komutunun tüm geçmişi listelediğini gözlemleyin.

```
git clone git://github.com/git/hello-world.git
cd hello-world
gitk
```

6. git-daemon paketi kurulumu sonrası:

Servis yöneticisinden git\_daemon başlatın, aşağıdaki komut ile başlatıldığına emin olun:

```
# service git_daemon status
```

7. 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
```

8. subversion paketi kurulumu sonrası:

test dizininin sorunsuz bir şekilde eklenmiş olduğunu gözlemleyin:

```
# svn co http://svn.pardus.org.tr/uludag/trunk/doc/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
```

## 6 Environment alt bileşeni

1. lazarus paketi kurulumu sonrası:

Kmenüden programın sorunsuz açıldığını gözlemleyin.

2. 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
```

3. 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.

4. ipython-gui paketi kurulumu sonrası:

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

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

5. 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")
```

6. drscheme paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde çalıştığını gözlemleyin.

7. qt-creator paketi kurulumu sonrası:

Kmenüden uygulamayı açın ve sorunsuz bir şekilde açıldığını gözlemleyin.

## 7 Language alt bileşeni

### 7.1 tcl alt bileşeni

1. tcl, tcltk paketleri kurulumu sonrası:

Aşağıda bulunan komutun bir gui ekranı çıkardığını gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/test_tcl.tcl
wish test_tcl.tcl
```

### 7.2 Php alt bileşeni

1. php-cli, php-common, mod\_php paketleri kurulumu sonrası:

Servis yöneticisinden apache servisini başlatın.

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
```

### 7.3 Perl alt bileşeni

1. perl-libwww paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/libwww-perl.tar.gz
```

Konsoldan;

```
# cd libwww-perl
# perl t/TEST
```

Komutlarını çalıştırın ve testlerden "ok" sonuçlarının döndüğünü gözlemleyin.

2. perl-Yaml paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırın ve sorunsuz bir şekilde çalıştığını gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/perl-yaml-test.pm
perl perl-yaml-test.pm
```

3. perl-TimeDate paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/TimeDate.tar.gz
```

Konsoldan;

```
# cd TimeDate
# /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.

4. perl-Storable paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/Storable.tar.gz
```

Konsoldan;

```
# cd St
```

Komutlarını çalıştırın ve testlerden "ok" sonuçlarının döndüğünü gözlemleyin.

5. perl-RPC-XML paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/RPC-XML.tar.gz
```

Konsoldan;

```
# cd RPC-XML
# /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.

6. perl-IP-Country paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/IP-Country.tar.gz
```

Konsoldan;

```
# cd IP-Country
# /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.

7. perl-HTML-Parser paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/HTML-Parser.tar.gz
```

Konsoldan;

```
# cd HTML-Parser
# sudo /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.

8. Aşağıda bulunan paketler sadece kurulum testine tabidir.

perl-Git

9. perl-Date-Calc paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/Date-Calc.tar.gz
```

Konsoldan;

```
# cd Date-Calc
# /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.

10. perl-Class-ISA paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/Class-ISA.tar.gz
```

Konsoldan;

```
# cd Class-ISA
# /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.

11. perl-Class-Accessor paketi kurulumu sonrası:

Aşağıda bulunan dosyayı indirin ve açın.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/perl/Class-Accessor.tar.gz
```

Konsoldan;

```
# cd Class-Accessor
# /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.

12. 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.

13. perl-Compress-Zlib paketi kurulumu sonrası:

programming-tr.pdf git testini gerçekleştirin.

14. 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.

15. 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.

## 7.4 Python alt bileşeni

1. Aşağıda bulunan paketler sadece kurulum testine tabidir.

```
python-docs
gnome-python-docs
PyQt-doc
```

2. setuptools paketi kurulumu sonrası:

```
svn co http://svn.pardus.org.tr/uludag/trunk/kde/boot-manager/manager/
cd manager
sudo python setup.py build
sudo python setup.py install
```

3. django-tagging paketi kurulumu sonrası:

```
mkdir test
cd test
django-admin.py startproject test_django_tagging
cd test_django_tagging
ipython
import tagging
```

4. pycha paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pycha
import chavier
```

5. python-xlrd paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import xlrd
```

6. pytz paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pytz
```

7. python-Jinja2 paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import jinja2
```

8. python-sphinx paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import sphinx
```

9. mysql-python paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import MySQLdb
```

10. python-markdown paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import markdown
```

11. python-preprocess paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import preprocess
```

12. python-avahi paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import avahi
```

13. python-tornado paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import tornado
```

14. pyorbit paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import ORBit
```

15. gnome-python paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import bonobo
```

16. PyXML paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import xml
```

17. sip paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import sip
```

18. pysqlite paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import sqlite3
```

19. PyQt paketi kurulumu sonrası:

Aşağıdaki komutları çalıştırın ve bir pencerenin açıldığını gözlemleyin.

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

20. sip paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import sip
```

21. python-configobj paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import configobj
```

22. python-decorator paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import decorator
```

23. python-mako paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import mako
```

24. python-FormEncode paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import formencode
```

25. cheetah paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import Cheetah
```

26. SQLAlchemy paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:



```
# ipython
import sqlobject
```

27. PyOpenGL paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import OpenGL
```

28. imdbpy paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import imdb
```

29. mutagen paketi kurulumu sonrası:

Aşağıda bulunan komutun sorunsuz bir şekilde müzik dosyası etiklerini listelediğini gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/multimedia/sound/sound/sample.mp3
mutagen-inspect sample.mp3
```

30. python-libcap-ng paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import capng
```

31. python-twitter paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import twitter
```

32. python-cx\_freeze paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import cx_Freeze
```

Aşağıda bulunan komutları çalıştırın ve küçük bir pencerenin açıldığını gözlemleyin.

```
cd /usr/lib/python2.6/site-packages/cx_Freeze/samples/wx
sudo python setup.py build
sudo python setup.py install
python wxapp.py
```

33. pybluez paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import bluetooth
```

Eğer bluetooth aygıtınız var ise aşağıda bulunan komutları çalıştırın ve hatasız bir şekilde çalıştığını gözlemleyiniz.

```
# cd /usr/share/doc/pybluez/examples/advanced
# python inquiry-with-rssi.py
```

34. pylirc paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pylirc
```

Eğer infrared bir kumandanız var ise bu kumandayı aktive ettikten sonra aşağıda bulunan testi çalıştırınız ve hatasız bir şekilde çalıştığını gözlemleyiniz.

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

35. pygobject ve pygobject-docs paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import codegen
import gio
import glib
```

36. pygame ve pygame-doc paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pygame
```

37. gst-python paketi kurulumu sonrası: ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pygst
```

38. pyFltk paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import fltk
```

39. python-turboflot paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import turboflot
```

40. python-ldap paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import ldap
```

41. python-iptables paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import iptables
```

42. numpy paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırdığınızda testlerden ok sonucu döndüğünü gözlemleyin.

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

43. sympy paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırdığınızda testlerden ok sonucu döndüğünü gözlemleyin.

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

44. 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()"
```

45. PyXML paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import xml
```

46. PyX paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pyx
```

47. pyNotifier paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import pynotify
```

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

```
# ipython
import httpLib2
```

49. Django paketi kurulumu sonrası:

- Aşağıda bulunan komutu çalıştırın:

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

test adında bir dizinin oluşturulduğunu ve bu dizin altında aşağıda bulunan dosyaların oluşturulduğunu 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 oluşturulduğunu gözlemleyin:

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

- Ve dizin içerisinde aşağıdaki dosyaların oluşturulduğunu gözlemleyin:

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

50. python-memcached paketi kurulumu sonrası:

memcached 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
```

51. pygtk, pygtk-docs ve pygtk-demo paketi kurulumu sonrası:

Aşağıda bulunan komutu çalıştırın ve çıkan listeden bir uygulama seçip çift tıklayın ve sorunsuz bir şekilde çalıştığını gözlemleyin.

```
pygtk-demo
```

52. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import mpmath
```

53. mpmath paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import mpmath
```

54. python-M2Crypto paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import M2Crypto
```

55. winpdb paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import winpdb
```

(DeprecationWarning önemli değildir.)

56. cython paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import cython
```

57. lxml paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import lxml
```

58. python-RuleDispatch paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython  
import dispatch
```

59. python-nose paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import nose
```

60. PyICU paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import PyICU
```

61. python-simplejson paketi kurulumu sonrası:

ipython paketini kurun ve aşağıda bulunan komutları çalıştırın:

```
# ipython
import simplejson
```

## 7.5 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
```

2. libmatthew, slf4j paketleri kurulumu sonrası:

office-tr.pdf zemberek-openoffice testini gerçekleştirin.

## 7.6 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
```

## 7.7 Dotnet alt bileşeni

1. Aşağıda bulunan paketler sadece kurulum testine tabidir

```
taglib-sharp
ndesk-dbus
ndesk-dbus-glib
```

2. boo paketi kurulumu sonrası:

Aşağıda bulunan komutların sorunsuz çalıştığını gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/dotnet/boo-test.boo
booi boo-test.boo
```

3. 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
```

4. 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
# mono test_mono.exe
```

5. libgdiplus paketi kurulumu sonrası:

programming-tr.pdf mono testini gerçekleştiriniz.

- ocaml paketi kurulumu sonrası:

Aşağıda bulunan komutun sorunsuz bir şekilde çalıştığını gözlemleyin.

```
wget http://cekirdek.pardus.org.tr/~semen/dist/test/programming/language/ocaml-test.ml
ocaml ocaml-test.ml
```

- 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
```