

Yeni Başlayanlar İçin IRAF Kullanım El Kitabı IRAF 2.10 Sürümü

Yard.Doç.Dr. S.Gülseçen

Yard.Doç.Dr. H.Gülseçen

İ.Ü. Fen Fakültesi Astronomi ve Uzay Bilimleri Bölümü

National Optical Astronomy Observatories

(Operated by the Association of Universities for research in Astronomy, Inc. under cooperative agreement with the National Science Foundation)

Tucson, Arizona 85726,USA

Temmuz 2000

ÖNSÖZ IRAF genel amaçlı bir veri indirmeye ve analiz etme aracıdır ve UNIX benzeri bir çalışma yöntemi vardır. IRAF'ın kendi programlama dili (CL) olmasına rağmen, FORTRAN ve C dili ile geliştirilmiş programlar da IRAF'ta çalıştırılabilir.

Bu kullanım kılavuzu her ne kadar IRAF'a yeni başlayanlar için hazırlandı ise de her seviyedeki IRAF kullanıcısı için bir "başvuru kılavuzu" niteliği taşımaktadır.

Teknik içeriği olmayan kısa bir giriş bölümünden sonra temel IRAF işlemlerinin anlatıldığı bölüm yer alır. Bu bölümde, paket programların çalıştırılması, yardım alma, dizin kullanımı, parametre dosyaları ve parametreler ile ilgili işlemler, taskların çalıştırılması, data dosyaları ve görüntü (*image*) dosyaları konularına değinilmiştir. Data üzerinde yapılan işlemler bölümünde teypten data okuma ve teybe data yazma konularına ve bazı data formatlarına değinilmiş ve örnekler verilmiştir. IRAF'ta grafik işlemleri bölümünden sonra son bölüm olan görüntüler ile ilgili işlemler ve komutlar yer almaktadır. Ekler bölümünde önce IRAF ve NOAO paket programları ve tasklarına yer verilmiştir. IRAF Unix tabanlı bir yazılım sistemi olduğundan, kullanıcıya faydalı olacağı düşünülen "Temel UNIX Kavramları ve Komutları" ve "vi Editörü Komutları" bölümleri de eklenmiştir.

Kullanım kılavuzu hazırlanırken çeşitli kaynaklara başvurulmuş ancak Jeannette Barnes'ın "A Beginner's Guide to Using IRAF – IRAF Version 2.10" isimli çalışması esas alınmıştır. Bu çalışma IRAF Sürüm 2.10 için hazırlandığından, IRAF Sürüm 2.11'deki bazı farklılıklar da yansıtılmaya çalışılmıştır.

Kullanıcıların temel bilgisayar kullanımı ve programlama bilgisine sahip oldukları varsayılmıştır. UNIX işletim sistemi ile ilgili daha fazla bilgi için çeşitli UNIX (ve Linux) kaynaklarına başvurulabileceği gibi, S. Gülseçen'in "UNIX İşletim Sistemine Giriş: HP-UX altında Uygulamalar" isimli çalışmasına da bakılabilir. Esas alınan kaynakta, VMS işletim sistemi ile ilgili örnekler de olmasına rağmen bunlara bu çalışmada yer verilmemiştir.

Bazı terimler, hem bilgisayar hem de astronomi kaynaklarına özgün (ingilizce) biçimleri ile yerleştikleri için bu kılavuzda da bu şekilde kullanılmıştır. Bunlardan bazıları şöyle sıralanabilir: binary (ikili), task, text (metin), header (başlık), data (veri), pixel, cursor (imleç), frame.

Kullanım kılavuzunda yer alan örnekler Sun Ultra 2/2000 (200MHz Ultra Sparc, 512 MB RAM, 70 GB HD, Sun Solaris CDE ver. 1.2) bilgisayarında test edilmiştir.

İÇERİK

1. Giriş;

- 1.1. IRAF Nedir?
- 1.2. IRAF'ın Kullanıcı Bakış Açısından Yapısı
- 1.3. IRAF Nasıl Çalıştırılır ve IRAF'tan Nasıl Çıkılır?
- 1.4. IRAF Paket Programları ve Taskları

2. IRAF'ta Temel İşlemler

- 2.1. IRAF Paket Programlarının Çalıştırılması
- 2.2. IRAF'ta Yardım Almak
 - 2.2.1. Basit İşlemler
 - 2.2.2. On-Line Elkitabı Sayfalar
 - 2.2.3. HELP Taskı ile Yardım Almak
 - 2.2.4. PHELP Taskı ile Yardım Almak
 - 2.2.5. On-Line Yardım Sayfalarının Çıktısını Almak
 - 2.2.6. Belirli Bir İşlevi Gerçekleştiren Taskı Bulmak
- 2.3. IRAF'ta Dizin Kullanımı
 - 2.3.1. Dizin Yaratmak ve Dizin Silmek
 - 2.3.2. Dizin Değiştirmek
- 2.4. Parametre Dosyaları ve Parametreler ile İlgili İşlemler
 - 2.4.1. Parametreleri Listelemek
 - 2.4.2. Parametrelere Eski Değerlerini Atamak
- 2.5. Taskların Çalıştırılması
 - 2.5.1. Genel Yazım Kuralları
 - 2.5.2. Taskları arka Planda Çalıştırmak
 - 2.5.3. Taskların Çalışmasını Kesmek
- 2.6. Data dosyaları
- 2.7. IRAF Görüntü (Image) Dosyaları
 - 2.7.1. IRAF Header Dosyaları
 - 2.7.2. IRAF Pixel Dosyaları
 - 2.7.3. Pixel Data Tipleri
 - 2.7.4. Görüntünün Boyutu
 - 2.7.5. Genel Görüntü araçları

3. IRAF'ta Datayı Okumak ve Yazmak

- 3.1. Genel Teyp (Tape) İşlemleri
- 3.2. Teyp Ortamına Yazmak ve Teyp Bilgi yoğunluğunu Ayarlamak
- 3.3. FITS Formatlı Veriler Üzerinde Bazı Basit İşlemler
 - 3.3.1. FITS Görüntülerini Okumak
 - 3.3.2. FITS Görüntülerini Yazmak
- 3.4. Text (Metin) Dosyalar
- 3.5. Standart Olmayan Data Formatları
- 3.6. TAR Dosya Formatı
- 3.7. Bazı Ek G/Ç (I/O) Taskları

4. IRAF'ta Grafik

- 4.1. Etkileşimli Grafik Cursor Modu
- 4.2. Tek Boyutlu Grafik Taskları
- 4.3. İki Boyutlu Grafik Taskları
- 4.4. Grafik Metakod Dosyaları ile İlgili Tasklar

5. Görüntüler ile İlgili İşlemler ve Komutlar

5.1. Görüntüler ile İlgili İşlemler

5.2. Görüntü Ortamında Etkileşimli Cursor Kullanımı

5.2.1. Görüntü Ortamında Cursor Konumlarının Belirlenmesi

5.2.2. Görüntüleri İncelemek ve Düzeltmek

Ek A IRAF Paket Programları

Ek B NOAO Paket Programları

Ek C Bazı Temel UNIX Kavramları ve Komutları

Ek D vi Editörü Komutları

Kaynaklar

Acknowledgments

Teşekkür

1.Giriş;

1.1.IRAF Nedir

IRAF (Image Reduction and Analysis Facility) National Optical Astronomy Observatories (Tucson, Cerro Tololo) tarafından geliştirilen bir veri indirgeme ve analiz sistemidir. IRAF'ın, çok çeşitli cihazları destekleyen bir grafik sistemi vardır ve görüntüleme araçları olarak XIMTOOL (Sunview) ve SAOImage'i (X-Windows) destekler. IRAF, network ortamında da kullanılabilir.

Diğer sistemler – STSDAS (Space Telescope Science Institute), PROS (ROSAT X-ray Observatory) ve IUE veri indirgeme ve analizi sistemi - , IRAF temel alınarak geliştirilmiştir ve böylece IRAF ile kımızı ötesi datadan X-ışın datusına kadar tüm data çeşitleri üzerinde işlem yapılabilir.

IRAF'ın adı ilk olarak 1981 yılında telaffuz edilmeye başlandı ve bugün IRAF, içinde yüzlerce uygulamayı barındıran ve çok çeşitli işletim sistemi platformlarında çalışabilen bir sistemdir. Gerek ABD'de gerek diğer ülkelerde pek çok kurum, proje ve kişi bu yirmi yıllık süre içerisinde IRAF için yazılım geliştirmiştir.

Günümüzde yarım düzine kadar büyük grubun yanı sıra, IRAF için yazılım geliştiren küçük gruplar ve kişiler vardır. Bunların çalışmaları ve IRAF'ın bilimsel açıdan gelişimi "IRAF User's Commitee" isimli bir komite tarafından izlenir ve yönlendirilir.

IRAF'ın ömrünü uzatabilmek için bilgisayar dünyasındaki her gelişme sisteme yansıtılmaya çalışılmaktadır ve böylece IRAF sürekli gelişmektedir.

1.2. Kullanıcı Bakış Açısından IRAF

IRAF'ın programlama dili olan CL (Command Language), IRAF'ı oluşturan sistem ve uygulama programlarını belli bir hiyerarşi içerisinde düzenler. Sistem ve uygulama programlarının her birine task (belirli bir işi yapan program), benzer işleri yapan programların oluşturduğu programlar grubuna da paket (*package*) adı verilmiştir. Metnin bundan sonraki bölümlerinde bu iki terim kullanılacaktır.

IRAF'ta programlar genellikle yaptıklarını çağrıştıran kelimelerle isimlendirilir. Örneğin FITS formatındaki dataları okuma taskı RFITS (ReadFITS), görüntüler ile ilgili taskların (HEDIT, IMARITH, IMDELETE v.b) toplandığı paket program IMAGES adını alır.

Tasklar ve paketler, IRAF komut satırından (bkz. Bölüm 2.2)icra edilir ve bir paket program bir ya da birden fazla iç içe alt programdan oluşur. Bu hiyerarşik yapı, ana dizin-alt dizin (*directory-subdirectory*) yapısının benzeridir; bazı programları çalıştırmak için belli bir yol (*path*) izleyerek onların bulunduğu yere gidilmeli, bazı programlar ise her yerden çalıştırılabilir.

1.3. IRAF Nasıl Çalıştırılır ve IRAF'tan Nasıl Çıkılır?

IRAF'ı çalıştırmak için kullanıcının “login” ya da “home” dizinine geçmesi ve **cl** komutunu yazması gerekir. UNIX ortamından “cl” komutu ile çalıştırılır. Eğer işistasyonu ortamında çalışılıyorsa tabi ki önce *gterm* veya *xterm* pencereleri açılmalıdır. Programdan **logout** komutu ile çıkılır. Bu komut uygulanmadan pencerenin kapatılması sakıncalıdır.

1.4. IRAF Paket Programları ve Taskları

IRAF sistemi iki büyük paket program grubundan oluşmuştur: sistem paket programları ile data indirgeme ve analiz etme paket programları.

(NOT: çeşitli dokümanlarda birinci grup IRAF Paket Programları (IRAF packages), ikinci grup da NOAO Paket Programları (NOAO packages) olarak geçer. Okuyucuya kolaylık sağlaması için biz de bundan sonra bu terimleri kullanacağız.)

IRAF paket programları IRAF'ın çekirdeğini (*core*) oluşturur. NOAO paket programları ise optik astronomi dataları üzerinde fotometri ve spektroskopi amaçlı indirgeme ve analiz işlemlerini yapan programlar grubudur. Her iki gruptaki programlar ve yaptıkları iş sırası ile **şekil 2** ve **şekil 3**'te yer almaktadır. Paket programlar ve bunların içinde bulunan alt programların tam listesi ise Ek A ve Ek B'de yer almaktadır.

dataio	data formatını dönüştürme (RFITS vb.)
dbms	data tabanı yönetimi (geliştirilme aşamasında)
images	genel görüntü işleme
language	IRAF komut dili
lists	liste işleme
nlocal	çeşitli kurumların kendi amaçları için geliştirdiği
tasklar	
obsolete	eski IRAF sürümlerinde geçerli olan tasklar
plot	grafik ve çizim
proto	yeni geliştirilen tasklar
softtools	yazılım araçları (programlama ve sistem bakım)
tv	görüntülerin yüklenmesi ve kontrolü
system	sistem hizmet programları (dosya işlemleri vb.)
utilities	çeşitli hizmetler
imcnv	farklı formatları IRAF görüntü formatına dönüştüren ve tersi işlemleri yapan tasklar

Şekil 1:IRAF sistem paket programları (Sürüm 2.11'e göre uyarlanmıştır)

artdata	yapay data üretimi
astrometry	astrometri
astutil	astronomi ile ilgili çeşitli işlemler
digiphot	dijital fotometri
focas	sönük görüntü sınıflandırma ve analizi
imred	görüntü indirgeme
mtlocal	özel NOAO format teypleri için giriş/çıkış işlemleri
nobsolete	obsolete tasks to be phased out in a future release
nproto	prototip işlemleri
observatory	gözlemevi ile ilgili parametreleri inceleme ve tanımlama
onedspec	tek boyutlu spektrum indirgeme ve analizi
rv	radyal hız analizi
twodspec	iki boyutlu spektrum indirgeme ve analizi
surfphot	galaksi izofot analizi

Şekil 3:Data indirgeme ve analizi paket programları (NOAO paket programları)

2.IRAF'ta Temel İşlemler

2.1. IRAF Paket Programlarının Çalıştırılması

IRAF ile bir işi yapabilmek için o işin tanımlandığı alt programa, 1. Bölümde de bahsedildiği gibi, adım adım ulaşmak gerekir. Bunu yapabilmek için de söz konusu alt programın hangi paket programda olduğu bilinmelidir. Örneğin IRAF'a girer girmez, IRAF komut satırından CCDLIST taskını çalıştırmak istediğimizde aşağıdaki çıktıyı elde edeceğiz:

```
cl> ccdlist
ccdlist
ERROR: syntax error
```

Böyle bir durumu önlemek için önce CCDLIST taskının nerede olduğuna bakmalıyız.

“cl> **help ccdlist**” komutunun icra edilmesi ile taska ilişkin bir yardım sayfası görüntülenecektir. Bu bilgilerden CCDLIST'in, NOAO paketinin içindeki IMRED taskının içindeki CCDRED taskının içinde olduğunu anlarız (bkz. Ek B). Bir kez taskın yerini bulduktan sonra onu çalıştırmak için aşağıdaki yol izlenir:

```
cl> noao
no> imred
im> ccdred
cc> ccdlist
```

Bir program, adı veya adını temsil eden yeteri kadar karakteri IRAF komut satırında yazmak ve Enter tuşuna basmak sureti ile çalıştırılır. Bu yöntemle, istenilen programa ulaşılan ve program çalıştırılana kadar devam edilir. Her aşamada o anda ulaşılan programın ileti işareti (prompt) görüntülenir; **bye** komutu ile, en son çalıştırılan programdan başlayarak adım adım geri dönülür ve cl> iletilisine ulaşılır. Şimdi aşağıdaki uygulamayı yapalım:

```
cl> package(paket programların listesini verir)
cl> softools(SOFTTOOLS programını çalıştırır)
```

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

```
so> package(SOFTTOOLS programının altprogramlarını listeler)
so> bye(bir üst seviyedeki programa dönlür)
cl>
```

IRAF'a girince, IRAF çekirdeğinde bulunan tüm paket programlar otomatik olarak yüklenir ve yukarıdaki örnekten görüldüğü gibi bulunduğumuz yerden hangi programların yüklenmiş olduğunu kontrol edebiliriz.

2.2. IRAF'ta Yardım Almak

IRAF'ta kullanıcıya yardımcı olacak ve kolaylık sağlayacak bazı işlemler vardır.

(NOT: Bundan sonra yer alan örneklerde çeşitli paket ve taskları çalıştırdığımızda o taska ait iletiyi değil hep cl> iletisini kullanacağız. Ancak kullanıcı uygulamayı bilgisayar başında yaptığında ilgili paket veya taskın iletisi ile karşı karşıya geleceğini unutmamalıdır.)

2.2.1. Basit İşlemler

```
cl> ?(içinde bulunulan paketin tasklarını listeler)
cl> ?dataio(belirli bir paketteki taskları listeler – paket yüklenmiş olmalı)
cl> ??(yüklenmiş olan tüm taskları listeler)
cl> package(yüklenmiş; olan tüm paketleri listeler)
```

2.2.2. Online “elkitabı” Sayfaları

IRAF sistemindeki tüm tasklara ait ve “elektronik elkitabı” diyebileceğimiz online yardım sayfaları vardır. Ayrıca her paket programa ait ve içinde o paketin içerdiği tüm taskları ve her taska ilişkin bir satırlık bir açıklamanın yer aldığı online menü sayfaları da vardır. Bunlar “IRAF Online Yardım Veri Tabanı” (“*IRAF Online Help Database*”)’nın bir parçasıdır. Bu veri tabanı dinamiktir; IRAF çalışır çalışmaz aktif olur ve bir IRAF oturumu boyunca geçerlidir. Buradan yardım alabilmek için, yardım alınmak istenen programa girmiş olmak gerekmez.

2.2.3. HELP Taskı ile Yardım Almak

HELP taskı belirli bir task veya paket için ekrana bilgi getirir. Eğer gelen bilgi bir sayfayı aşyorsa veya bu ortamdan çıkılmak isteniyorsa aşağıdaki tuşlar kullanılabilir:

```
(enter)sayfayı bir satır ilerletir
(d)sayfayı yarım sayfa ilerletir
(ara çubuğu)sayfayı bir sayfa ilerletir
(q)cl> ortamına geri götürür
```

Şimdi birkaç basit uygulama yapalım:

```
cl> help(o andaki paket program için açıklama)
cl> help help(HELP taskı için açıklama)
cl> help rfits sec=examples(RFITS'in EXAMPLES bölümü için aşağıdaki gibi açıklama getirir)
```

1. Convert all the image data on a mag tape to individual IRAF images. Allow rfits to select the output datatype and set blanks to zero.
cl> rfits mtbl600 ' ' images or alternatively
cl> rfits mtbl600 * images
2. Converts FITS files on disc to.....

cl> **help rfits opt=source**(RFITS taskının kaynak kodunu gösterir)

cl> **help dataio**(belirli bir paket (şu anda DATAIO) içindeki tasklarla ilgili kısa açıklama)

2.2.4. PHELP Taskı ile Yardım Almak

Bu task HELP taskının benzeridir ancak kullanıcının ekrana gelen on-line yardım sayfalarında ileri ve geri gitmesini sağlar ve pek çok seçeneğe sahiptir. PHELP taskını birkaç örnek üzerinde inceleyelim:

```
cl> phelp phelp(PHELP'in kendisi için bilgi getirir)
      ?(ekrana bir satırlık özet bilgi yazar)
      q(bulunulan ortamdan çıkılmasını sağlar)
cl> phelp combine all+("combine" adını taşıyan tüm
      tasklar hakkında bilgi getirir)
      N(bir sonraki "combine" taskı ile ilgili
      bilginin bulunduğu sayfaya gider)
      P(bir önceki "combine" taskı ile ilgili
      bilginin bulunduğu sayfaya gider)
      q (bulunulan ortamdan çıkılmasını sağlar)
```

2.2.5. On-line Yardım Sayfalarının Çıktısını Almak

On-line yardım sayfalarının veya herhangi bir *.hlp* uzantılı dosyanın çıktısı HELP ve LPRINT taskları ve (|) yönlendirme işareti (*pipe*) kullanılarak yapılır. Başlangıçta tanımlanan yazıcıdan farklı bir yazıcı kullanılacaksa bunun *set* komutu tanımlanması gerekir. Aşağıdaki örnekleri inceleyiniz:

```
cl> help help | lprint(yardım alma ile ilgili bilgilerin bulunduğu dosyayı yazıcıya gönderir)
cl> help help | lprint dev=<lw>(yardım alma ile ilgili bilgilerin bulunduğu dosyayı belirli
bir yazıcıya gönderir)
cl> help dataio.* | lprint(DATAIO paketi ile ilgili tüm bilgiyi yazıcıya gönderir)
```

2.2.6. Belirli bir İşlevi Gerçekleştiren Taskı Bulmak

Belirli bir işlevi gerçekleştiren tüm tasklar REFERENCES taskı ile listelenebilir. Bunu için on-line yardım veri tabanında arama yapılır ve kullanıcının yazdığı ifade ile uyuşan tüm taskların listesi ekrana getirilir. Aşağıdaki uygulamaları inceleyiniz.

```
cl> references help(tanımlama bölümlerinde "help" ifadesini
      içeren tüm tasklar listelenir)
cl> refer upd+(yukarıdaki örnekten de anlaşılacağı gibi bilgiler
      oldukça yavaş gelmekte; bundan sonraki aramaları
      hızlandırmak için bir hızlı-referans dosyası
      üretilir;
      bunun etkisini takip eden örnekte göreceksiniz)
cl> refer smooth(hızlı-referans dosyası kullanılarak, tanımlama
      bölümlerinde "smooth" ifadesini içeren tüm tasklar
      listelenir)
```

2.3. IRAF'ta Dizin Kullanımı

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

IRAF, görüntü ya da sanal (*virtual*) dosya sistemi kullanır. Bunun kullanıcı için anlamı şudur: IRAF hangi bilgisayarda kullanılırsa kullanılsın izin ve dosya isimleri aynıdır.

Dizin kullanımını IRAF'ta çok önemlidir. Sistemi ancak içinde bulunduğu dizinden (home directory) çalıştırabileceğinizi unutmayınız! Aşağıdaki iki uygulama neyi göstermektedir?

```
cl> show home
cl> show uparm
```

2.3.1. Dizin Yaratmak ve Dizin Silmek

IRAF'ta izin MKDIR taskı ile yaratılır. Aşağıdaki örneklerde hem IRAF'ın içinde bulunduğu dizinde hem de daha üst dizinlerde alt dizinler yaratılabileceğini göreceksiniz.

```
cl> diskpace(disk üzerindeki boş alan kontrol edilir)
cl> mkdir nite1(nite1 isimli bir alt izin yaratılır)
cl> mkdir home$nite1(IRAF'ın içinde bulunduğu dizinde nite1 isimli bir alt izin yaratır)
cl> mkdir /tmp8/irafdir(tmp8 isimli dizinde irafdir dizinini yaratır)
```

IRAF'ın içinden izin silmek mümkün değildir. Bu ancak işletim sistemi (UNIX) seviyesinden yapılabilir; ancak (!) karakteri kullanılarak IRAF'ı terk etmeden de sanki işletim sistemi seviyesinde bulunuluyormuş gibi izin silinebilir. Örnekleri inceleyiniz.

```
cl> cd nite1(nite1 dizinine geçilir)
cl> imdelete *.imh(nite1 dizinindeki tüm görüntü dosyaları silinir) cl> cd ..(bir üst dizine geçilir)
cl> !rm -r nite1(IRAF'tan çıkmadan, işletim sistemi seviyesindeymiş gibi, nite1 dizini silinir)
```

2.3.2. Dizin Değiştirmek

Kullanıcı IRAF'ın içinde olsa da, IRAF izin isimlerini, mantıksal izin isimlerini ve işletim sistemi (UNIX) yol isimlerini kullanarak dizinler arasında dolaşabilir.

```
cl> path(içinde bulunulan izin adını verir)
cl> cd(IRAF'ın içinde bulunduğu dizine geçilir)
cl> mkdir nite1(nite1 isimli bir alt izin yaratılır)
cl> cd nite1(nite1 alt dizinine geçilir)
cl> cd ..(bir üst dizine çıkılır)
cl> cd iraf$doc(iraf mantıksal dizininin doc isimli alt dizinine geçilir)
cl> back(bir önceki dizine döner)
```

2.4. Parametre Dosyaları ve Parametreler ile İlgili İşlemler

IRAF'ta tasklar bir takım parametreler ile çalıştırılır. Parametreler o taskın işlevini nasıl gerçekleştireceğini belirler. Bu bölümde parametrelerin nasıl tanımlandığını ve gerektiğinde nasıl değiştirildiğini inceleyeceğiz. Aşağıda RFITS taskının parametreleri görülmektedir.

fits_file	=	'mta'	FITS data source
file_list	=		File list
iraf_file	=	' '	IRAF filename
(make_image	=	yes)	Create an IRAF image?
(long_header	=	no)	Print FITS header cards?

(short_header	=	yes)	Print short header?
(datatype	=	' '	IRAF data type
(blanc	=	0.	Blank value
(scale	=	yes)	Scale the data?
(oldirafname	=	no	Use old IRAF name in place of iraf_file?
(offset	=	0)	Tape file offset
(mode	=	'q1'	

İki tür parametre vardır: **sorgu** veya **konum parametreleri** (*query or positional parameters*) ile **saklı parametreler** (*hidden parameters*).

Sorgu parametreleri taskın her çalıştırılışında kesinlikle belirtilmelidir. Buna karşın saklı parametreler task çalışmaya başlamadan önce belirtilmedi ise eski değerleri ile geçerli olmaya devam edecektir. Kullanıcı, herhangi bir taskın parametre değerlerini yapacağı iş doğrultusunda kendisi için özelleştirebilir. Bu yeni değerleri içeren parametre dosyaları kullanıcının *uparm* isimli dizininde depolanır. Herhangi bir task çalıştırıldığında IRAF ilk önce bu dizine “bakar” ve buradaki değerleri dikkate alır. Eğer böyle bir dosya yoksa sistem tarafından tanımlanmış olan değerler kullanılır.

uparm dizininde yer alan parametre dosyalarının isimleri paket ve taskın isimlerinin belirli bir biçimde birleştirilmesi (paket adının ilk iki karakteri ile son karakteri ve task adının ilk beş karakteri ile son karakteri) ile oluşturulur. Örneğin **aptqphot** adı **apphot** ile **qphot** kelimelerinden türetilmiştir. Farklı örnekleri görmek için “dir uparm” komutunu işletiniz.

Paketlerin de parametre dosyaları olabilir. Bu durumda parametre değerleri o paketi oluşturan tüm tasklar tarafından kullanılır.

2.4.1. Parametreleri Listelemek ve Edit Etmek

Parametreler LPARAM taskı ile listelenebilir. Herhangi bir paketin parametrelerini listelemeden önce o paketin yüklenmiş olması gerekir. Eğer daha önceden kullanıcı bu paketin parametre dosyasında değişiklik yaptı ise dosya bu hali ile listelenecektir. Örnekte RFITS taskının parametreleri listelenecektir.

```
cl>lpar rfits
```

Parametreler üzerinde her türlü işlemi EPARAM (EditPARAMeters) taskı ile gerçekleştirebiliriz. EPARAM ile, *editor* çevre değişkeni tarafından kontrol edilen bir etkileşimli monitör editörü devreye sokulur. Önceki bölümlerden de hatırlayacağınız gibi IRAF’ta bir taskı ya da paketi yeteri kadar harf yazarak komut satırından çağırabiliyor ya da çalıştırabiliyorduk. Örneği inceleyiniz (eparame yerine epar):

```
cl> epar rfits(RFITS’in parametreleri üzerinde işlem yapabilmek için etkileşimli monitör editörünü açar)
```

Etkileşimli monitör editörünün basit bir kullanımı vardır. Cursor’ı istenilen parametreye konumlandırdıktan sonra, parametrenin yeni değeri girilir ve (Enter) tuşuna basılır. İşlem bitince :q komutu ile parametre ortamından çıkılır. Parametre dosyası yeni parametre değerleri ile *uparm* dizinine kaydedilmiş olur. Aşağıda hangi tuş/tuşlar yardımı ile parametre listesi üzerinde hareket edileceği ve istendiğinde o ortamdan çıkılabileceği açıklanmıştır.

(Enter) , ↓ , (Ctrl+J) cursor aşağıya doğru hareket eder
, (Ctrl+K) cursor yukarıya doğru hareket eder

(Ctrl+Z) veya (Ctrl+D) parametreler güncelleştirilir ve EPARAM'dan çıkılır
(Ctrl+C) EPARAM'dan parametreler güncelleştirilmeden çıkılır

Parametreleri değiştirmenin ikinci bir yolu da komut satırında task adını, parametreyi ve parametrenin yeni değerini açıkça yazmaktır. Örneklerde de görüleceği gibi çeşitli tipte parametre değerleri için çeşitli yazım biçimleri (sintaks) kullanılır: örneğin boolean değerler için **yes** ve **no** değerleri kullanılırken, karakter türündeki değerler “ “ işaretleri arasında belirtilmelidir.

```
cl> rfits.scale=no
cl> rfits.datatype="real"
cl> rfits.offset=100
cl> =rfits.datatype
```

2.4.2. Parametrelere Eski Değerlerini Atamak

Bazı durumlarda parametrelere IRAF'ın başlangıçta belirlediği (default) değerleri yeniden atamak gerekli olabilir. Bunun için UNLEARN komutu kullanılır. Bu komut ile tek bir taskın parametre değerlerine dönülebileceği gibi bir paketteki tüm tasklar için aynı şey yapılabilir.

```
cl> unlearn rfits (RFITS taskı için IRAF'ın belirlediği değerlere dönmek)
cl> unlearn dataio (DATAIO paketindeki tüm tasklarda ilk değerlere dönmek)
```

2.5. Taskların Çalıştırılması

2.5.1. Genel Yazım Kuralları

Önceki bölümlerden de hatırlanacağı gibi taskları çalıştırmak için IRAF komut satırından taskın adını ya da onu IRAF'a tanıttığımız sayıda karakteri yazmak ve (Enter) tuşuna basmak yeterliydi. Şimdi bununla ilgili birkaç farklı yolu inceleyeceğiz. Örneklerde IMSTATISTIC taskını kullanacağımız için önce bunun parametrelerini inceleyelim:

```
images = Images
(fields = "image,npix,mean,stddev,min,max") Fields to be
printed
(lower = INDEF) Lower cutoff for pixel values
(upper = INDEF) Upper cutoff for pixel values
(binwidth = 0.1) Bin width of histogram in sigma
(format = yes) Format output and print column
labels?
```

1. En basit yol, komut satırından taskın parametre listesini ekrana çağırdıktan ve sorgu parametrelerini (parantez içinde olmayanları) tespit ettikten sonra yine komut satırında taskın adını yazıp (Enter) tuşuna basmak ve otomatik olarak ekrana gelen parametre isimlerinin karşısına değerlerini girmektir. Saklı parametrelerin ilk değerleri geçerli olmaya devam eder.
2. cl> **imstat**
3. Images: **dev\$pix**
4. # IMAGES NPIX MEAN STDEV MIN MAX
5. dev\$pix 26214 108.3 131.3 -1. 19936.

Sorgu yöntemi ile saklı parametrelerin değerleri de değiştirilebilir (eğer değişmesi gerekiyorsa). Aşağıdaki örnekte `format` parametresinin değeri 'yes' iken 'no' olarak değiştirildi. '-' işaretinin etkisine dikkat edin! Boolean parametrelerin '=yes' ve '=no' bölümleri sırası ile '+' ve '-' işaretleri ile belirtilebilir.

```
cl> imstat form-
Images:      dev$pix
             dev$pix& 26214   108.3  131.3   -1. 19936.
```

6. İkinci bir yol, parametreleri ve yeni değerlerini komut satırında yazarak taskı çalıştırmak. Bu çalıştırma biçiminde gerekli parametreler komut satırında, parametre listesindeki sıralarına uygun olarak verilmelidir. Parametre adı yazılmasa da olur, sadece değerini yazmak yeterlidir.

Saklı parametreler de komut satırından tanımlanabilir. Bunların adı da değeri de komut satırında yer aldığı için, sorgu parametrelerinden sonra olmak kaydı ile herhangi bir sırada yazılabilir. Bu parametrelerin yeni değerleri taskın sadece bu icrası için geçerlidir; tasktan çıkılınca eski değerleri geçerli olmaya devam eder.

```
cl> imstat dev$pix fields=mean,stddev
```

Aşağıdaki örnekte, bir satırı aşan bir komut görüyorsunuz. İkinci satıra “\” işareti ile devam edilmiştir.

```
cl> imstat dev$pix fields=mean,stdev,min,max \>>> format-
```

7. Tabi ki bunun yerine her zaman daha pratik bir yol olan EPARAM taskını çalıştırarak da saklı parametrelerin değerleri değiştirilebilir ve çalışması istenen asıl task çalıştırılabilir. cl> **epar imstat** (bu komuttan sonra parametre değerleri değiştirilir ve aşağıdaki komut uygulanır)

```
cl> imstat dev$pix
```

8. cl> **EPARAM** “task adı” komutu ile ekrana gelen parametre listesinde parametre değerleri değiştirildikten sonra da gerekli task çalıştırılabilir. Bunu için parametre listesindeki değişiklik işi bitince **:go** komutu ile ortam terkedilir.

Her IRAF kullanıcısının tercih ettiği bir yol vardır ancak uygulamalar göstermiştir ki en çok kullanılan en son anlatılan yöntemdir.

2.5.2. Taskları Arka Planda Çalıştırmak

IRAF’ta taskları arka planda çalıştırmak mümkündür. Bunu komut satırında komutun sonuna **&** işaretini ilave ederek yapabiliriz. Arka plana atılmış işler IRAF’tan çıkılsa da çalışmaya devam eder.

Aşağıdaki örnekte, IMSTATISTICS taskı arka planda çalışmak üzere ayarlanmıştır. Ancak taskın üreteceği çıktı tanım gereği terminale geleceğinden, bunu önlemek için *statfile* isimli dosyaya yönlendirme yapılmıştır (**&** ile ilgili bilgi daha sonraki bölümlerde vardır).

```
cl> imstat dev$pix form- > statfile &
```

Yukarıdaki komutta gerekli parametrelerin eksik yazılması durumunda CL aşağıdaki örnekte olduğu gibi işlemi durdurup sizin değer girmenizi bekleyecektir.

```
cl> imstat >> statfile &
1 stopped waiting for parameter input
cl> service 1
Images: dev$pix
```

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

Böyle bir durumda, eğer kullanıcı hangi parametreden bahsedildiğinden emin değilse, yapılabilecek en iyi iş işlemi iptal edip (*kill*) yeniden başlatmaktır. JOBS taskı ile arka plana atılmış işler numaraları ile birlikte listelenebilir. Bunları iptal etmek için **kill** komutu ve iş numarası girilmelidir.

```
cl> jobs
cl> kill 1
```

2.5.3. Taskların Çalışmasını Kesmek

Bir taskı çalıştırdıktan sonra herhangi bir sebeple iptal etmek isteyebiliriz. Bunun için **Ctrl-C** kullanılır. Böyle bir işlemten sonra her şeyi eski haline getirmek için genellikle **flprcache** komutu kullanılır (bu komut uygulamada “flipper” olarak isimlendirilir). Bundan sonra da hala bazı problemler varsa bir kez daha aynı komut çalıştırılır ve problem devam ediyorsa yapılabilecek en iyi iş IRAF’tan çıkıp tekrar girmektir.

Genellikle her IRAF paketi (taskları ile birlikte) bir tek exe dosyası yaratılmıştır. Herhangi bir paketteki bir task çalıştırıldığında, o taskın içinde yer aldığı paketin tamamı kullanıcının “işlem belleği” ‘ne (“*process cache*”) yerleştirilir. Bu bellek alanı istendiğinde **prcache** komutu ile incelenebilir. **kill** komutu ile bir taskın çalışması kesildiğinde büyük bir olasılıkla paketin içinde bulunan exe dosyasında bozulma oluşur bu da bu dosyada bulunan başka bir taskın yanlış çalışmasına sebep olabilir. flpr, kullanıcının işlem belleğinden tüm exe dosyalarını (SYSTEMS hariç) atar ve burayı temizler.

Teyp işlemleri ile ilgili taskların çalışmasını kesmek tehlikelidir ve kullanıcının, çok zorunlu değilse, bunu yapmaması önerilir.

2.6. Data Dosyaları

Daha önceki bölümlerde de bahsedildiği gibi, IRAF görüntü (virtual) dosya sistemi kullanır: hangi işletim sisteminde kullanılırsa kullanılsın tüm IRAF dosya isimleri aynıdır. Geçerli dosya isimleri [**A-Za-z0-9_.**] karakter kümesindeki karakterler kullanılarak oluşturulabilir. Ayrıca “1245b” veya “03457” gibi değişik dosya isimlerini mutlaka “ “ işaretleri arasında yazmak gerekir (birincisi sekizli sayıları çağrıştırdığı ikincisi de 0 ile başladığı için).

IRAF data dosyaları üçe ayrılır:

1. Metin (ASCII) dosyalar
2. Basit ikili (binary) dosyalar
3. Görüntü (image) dosyalar.

Görüntü dosyalar özel bir dosya türü olduğundan daha sonra incelenecektir. Bu dosyalar ile çalışılacağı zaman IMAGES paketindeki tasklar kullanılır. Metin ve ikili dosya işlemleri için ise SYSTEMS paketindeki tasklar kullanılmalıdır.

DIRECTORY taskı ile o anda içinde bulunulan veya belirtilen bir dizindeki dosyalar listelenir. Aşağıdaki örnekleri inceleyiniz ve joker karakterlerinin kullanımına dikkat ediniz:

```
cl> dir          (bulunulan dizinin içeriğini listeler)
cl> dir uparm    (uparm dizinindeki dosyaların listesini verir)
cl> dir *cl*     (adında "cl" karakter dizisi olan dosyaları listeler)
cl> dir login.??(adı login ve uzantısı herhangi iki karakter olan
                dosyaları listeler)
```

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

```
cl> dir *.imh (bu dizindeki tüm görüntü dosyalarının listesini verir)
```

Metin dosyaları IRAF'ta çok farklı amaçlarla kullanılabilir; bazı taskların girdi bilgilerini içerirken bazı taskların da ürettiği bilgileri saklamak için kullanılabilir (çoğunlukla veri tabanı ve log dosyaları olarak). Bir taskın üreteceği terminal çıktısı bir metin dosyasına yönlendirilebilir. Bu tür dosyaların (uygun olursa ikili dosyaların da) içinde işlem yapmak için IRAF'ın EDIT komutu, dosyalarla işlem yapmak için ise SYSTEM paketindeki tasklar kullanılır.

concatenate	- birkaç dosyayı birleştirir
copy	- dosya kopyalar (görüntüler için IMCOPY)
count	- bir metin dosyasındaki satır, kelime ve karakter sayısını verir
delete	- dosya siler (görüntüler için IMDELETE)
head	- bir metin dosyasının ilk birkaç satırını yazar
lprint	- bir dosyayı yazıcıya gönderir
match	- belirli bir karakter dizisi ile uyuşan tüm satırları yazar
page	- bir dosyanın içinde hareketi sağlar
rename	- dosyanın yeniden adlandırılmasını sağlar
sort	- bir metin dosyasında sıralama yapar
tail	- bir dosyanın son birkaç satırını yazar
type	- bir dosyanın içeriğini standart çıktı cihazında yazar

LISTS paketindeki bazı tasklar ve PROTO paketindeki FIELDS ve JOINLINES taskları yardımı ile datalar üzerinde basit işlemler yapılabilir. NOAO paketi gibi diğer harici paketlerin ise metin dosyaları için özel taskları vardır.

2.7. IRAF Görüntü Dosyaları

IRAF birçok görüntü formatını desteklemektedir. Bu bölümde sadece IRAF'ın özgün formatı olan **OIF (Original IRAF Format)** hakkında bilgi verilecektir.

IRAF'ın kurulumu sırasında, değiştirilmedikçe, OIF varsayılan görüntü formatıdır. Görüntü formatını kontrol etmek ve **imtype** çevre değişkeni yardımı ile yeniden tanımlamak mümkündür. Örnekleri inceleyiniz:

```
cl> show imtype (görüntü formatını kontrol eder)
cl> set imtype=imh (görüntü formatını OIF olarak tanımlar)
```

Genel olarak söylemek gerekirse **imtype** değişkeni sadece RFITS veya başka bir okuyucu ile okunan yeni görüntüleri etkiler. Görüntü formatına bağlı olsa da çıktı olarak üretilen görüntüler girdi görüntülerinin formatında olur (eğer girdi görüntünün formatı haricen tanımlanmaz ise).

OIF formatı iki dosyadan oluşur: uzantısı **.imh** olan bir header dosyası ve uzantısı **.pix** olan bir pixel dosyası. UNIX-IRAF sistemlerinde her bir **.imh** uzantılı dosyaya karşılık bir de **“..”** dosyası yaratılır. Bu görüntünün DELETE komutu ile yanlışlıkla silinmesini önler (bu dosyalar **dir a+** komutu ile görülebilir).

2.7.1. IRAF Header Dosyaları

IRAF header (*başlık*) dosyası uzantısı **.imh** olan bir binary dosyadır. Bu dosyanın içeriği ve bununla ilgili parametreler IMHEADER taskı ile görülebilir. FITS header dosyaları gibi görünse de onlardan oldukça farklıdır.

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

```
cl> imheader dev$pix(görüntü header dosyasını kısaca gösterir)
cl> imheader dev$pix l+ | page(görüntü header dosyasının tamamını gösterir)
```

Görüntü header dosyasının *min_lenuserarea* çevre değişkeni ile kontrol edilen ve yeniden tanımlanabilen belirli bir uzunluğu vardır, bu da 20000 karakterdir (250 satır x 80 karakter).

```
cl> show min_lenuserarea(o andaki değeri gösterir)
cl> set min_lenuserarea = 40000(uzunluk yeniden tanımlanır)
```

HEDIT taskı ile header dosyasındaki bilgilere yeni anahtar kelimeler (keywords) eklenebilir veya mevcut olanlar üzerinde değişiklik yapılabilir. HESELECT ile de seçilmiş bazı anahtar kelimelerin değerleri listelenebilir. Aşağıdaki örnekleri uygulayınız:

```
cl> imcopy dev$pix newpix(dev$pix dosyasının bir kopyasını çıkarır)
cl> hedit newpix newkey 'string' add+(newkey isimli yeni bir anahtar kelime ekler)
cl> hselect newpix $I,newkey yes(newkey'in değerini yazar)
cl> imdelete newpix(1. Örnekte yaratılmış olan kopyayı siler)
```

2.7.2. IRAF Pixel Dosyaları

IRAF pixel dosyaları uzantısı *.pix* olan birer binary dosyadır. Pixel dosyaları header dosyaları ile aynı dizinde olmak zorunda değildir. IRAF görüntüleri yaratıldığında **imdir** çevre değişkeni pixel dosyalarının (*.pix*) nereye kaydedileceğini belirler. Bunlara ulaşılacak yol adı da header dosyasına (*.imh*) yazılır. Böylece görüntülerle ilgili bir işlem yapacak herhangi bir task o görüntü ile ilgili pixel dosyası nerede olursa olsun onu bulacaktır. Başlangıçta *.imh* dosyaları o anda içinde bulunulan dizine yerleşir. Aşağıdaki örnekleri incelerken kullanıcı şunu unutmamalıdır: “**set imdir**” komutu dizin yaratmaz; dizinler MKDIR komutu ile yaratılır (HDR\$pixels/ dizini hariç – bu IRAF tarafından otomatik olarak yaratılır).

```
cl> show imdir (.pix dosyalarının içinde bulunduğu dizini gösterir)
cl> dir imdir (dizindeki dosyaları listeler)
cl> set imdir=HDR$ (o anda içinde bulunulan dizini .pix dosyalarının
                  içinde bulunacağı dizin olarak tanımlar)
cl> set imdir=HDR$pixels/
cl> set imdir =/tmp8/irafdir/
```

Yukarıdaki örneklerde kullanılan “/” işaretinin önemini bir kez daha hatırlatmak isteriz!

HDR\$ ve HDR\$pixels/ mantıksal dizinleri ana işletim sisteminden bağımsız birer yol adıdır ve görüntülerin header dosyalarında da aynı biçimde yer alırlar. Diğer dizin yol adı örnekleri işletim sistemi bağımlıdır.

2.7.3. Pixel Data Tipleri

IRAF aşağıdaki pixel data tiplerini destekler:

- **Short** (kısa) (16 bit işaretli tamsayı)
- **Long** (uzun) (32 bitlik işaretli tamsayı)
- **Real** (gerçel) (32 bitlik kayan noktalı)
- **Double**(çift hassasiyetli gerçel) (64 bitlik kayan noktalı)

Kompleks sayılar için IRAF'ın desteği çok sınırlıdır. Bir görüntünün data tipini değiştirmek için CHPIXTYPE taskı kullanılır. Pixel data tipi IMHEADER taskı ile belirlenebilir.

```
cl> imhead dev$pix  
dev$pix[512,512] [short] : m51 B 600s
```

IRAF'ın sistem taskları girdi görüntüsünün pixel tipinde çıktı görüntüsü üretir. Eğer dönüşüm sırasında karmaşık hesaplamalar varsa bunlar tek veya çift duyarlıklı gerçel sayılarla yapılabilir aksi takdirde buna gerek yoktur; girdi görüntüsünün pixel tipinde yapılır. Bazı tasklar hesaplamaların ve çıktı görüntüsünün hangi data tipi ile gerçekleştirileceğini tanımlayan parametrelere sahiptir. Tamsayı tipinde datası olan kullanıcılar temkinli olmalıdır: eğer hard disk miktarı yeterliyse, görüntüyü işlemeye başlamadan önce tamsayıların gerçel sayılara dönüştürülmesi tavsiye edilir; dönüştürülmez ise datanın bir kısmı kesilip atılır veya anlamsız negatif sayılar şeklinde ortaya çıkar.

2.7.4. Görüntünün Boyutu

IRAF çok boyutlu görüntüleri destekler. Bir görüntünün büyüklüğü ve boyutu IMHEADER taskı ile belirlenebilir. En son örneğimizden de anlaşılacağı gibi dev\$pix adı ile tanımlı görüntü 2-d boyutundadır ve her bir boyutu 512 pixel içerir.

Tek boyutlu (**oned**) bir görüntü bir satırı veya bir sütünü olan iki boyutlu bir görüntü olarak temsil edilebileceği gibi, tek boyutlu bir vektör olarak da temsil edilebilir. NOAO spektroskopik paketlerinde tek boyutlu datalar için bu, çok yaygın bir formattır.

```
cl> imhead oned  
oned [820] [real] : FEIGE 98
```

IMSTACK taskı ile birden fazla görüntü yüksek boyutlu bir görüntü yığını şekline dönüştürülebilir (yeni görüntünün header dosyası listedeki birinci görüntünün header dosyasındaki bilgilerini alır). IMSLICE taskı ile girdi görüntüsünün boyutunun bir eksiği boyutunda görüntü üretilebilir.

Takip eden bölümlerde açıklanan “görüntü bölümü” (“image section”), IMCOPY taskı ile birlikte bir görüntünün boyutlarını düşürmede kullanılır. Aşağıdaki örnekte de görüldüğü gibi iki boyutlu olan görüntü tek boyutlu bir vektöre dönüşmekte.

```
cl> ihmead dev$pix  
dev$pix[512,512] [short] : m51 B 600s  
cl> imcopy dev$pix[*],20] oned  
cl> imhead oned  
oned [512] [short] : m51 B 600s
```

2.7.5. Genel Görüntü Araçları

Bu bölümde birkaç genel amaçlı görüntü aracına değinilecektir. Görüntülerle çalışırken, SYSTEMS paketinde yer alan metin ve ikili dosyalar için geçerli benzer araçları kullanmak yerine bunları kullanmak önemlidir çünkü bunlar daha karmaşık dosya yapıları (örneğin bir dosya yerine iki dosya) ile başa çıkabilir. Aşağıda bu araçlar ve yaptıkları iş yer almaktadır:

```
IMCOPY   giriş görüntüsünün yeni bir kopyasını yaratır  
IMRENAME o anda kullanılan görüntünün adını değiştirir  
IMDELETE görüntüyü siler
```

Aşağıdaki örnekleri inceleyiniz:

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

```
cl> imcopy dev$pix newpix(newpix adı ile görüntünün bir kopyasını yaratır)  
cl> imrename newpix newcop(newpix'in adını newcop olarak değiştirir)  
cl> imdelete newcop(newcop'u siler)
```

IMRENAME taskının bir kullanışlı özelliği daha vardır; pixel dosyalarını **imdir** çevre değişkeni ile belirlenmiş o anda geçerli dizine taşır. Bunu yapmak için header dosyalarının adını ya da yerini değiştirmeye gerek yoktur.

```
cl> set imdir =</tmp3/irafdir/>(imdir hazırlanır)  
cl> imrename *.imh .(pixel dosyalarını yeni dizine taşır)
```

Pixel değerleri LISTPIXELS taskı ile listelenebilir. Şimdi aşağıdaki örnekleri inceleyelim:

```
cl> listpix dev$pix[20:25,30:32](görüntünün bir bölümündeki pixel değerlerini listeler)  
cl> listpix dev$pix[20:25,30:32] | table(listeler ve yeniden formatlar)
```

İkinci örnekte LISTPIXELS taskının çıktısı TABLE taskına yönlendirilir; çıktı görüntülenmeden önce LISTS paketinde bulunan TABLE taskı tarafından yeniden formatlanacaktır.

Görüntüler üzerinde işlem yapan diğer tasklar IMAGES ve PROTO paketlerinde yer alır (bkz. Ek A-B).

3. IRAF'ta Datayı Okuma ve Yazma

IRAF'ta data okuma ve yazma DATAIO paketindeki tasklar yardımı ile yapılır. Ayrıca SYSTEMS, SOFTTOOLS ve PROTO paketlerinde yer alan bazı genel amaçlı tasklar da bunun için kullanılabilir. IRAF ile birlikte çalışan diğer paketlerin de data okuma ve yazma ile ilgili kendi taskları vardır. Örneğin NOAO'nun MTLOCAL paketi, STSDAS'ın FITSIO paketi ve SAO'nun geliştirdiği PROS, XRAY'in XDATAIO paketi.

3.1. Genel Teyp (Tape) İşlemleri

IRAF aşağıdaki teyp cihazlarını destekler:

- 9-izli teyp sürücü
- QIC sürücü (kartuş)
- Exabyte
- DAT (Digital Audio Tapes) sürücü

IRAF'ın kullandığı teyp cihazları *dev\$tapecap* isimli dosyada tanımlıdır. Bu dosyada her cihazın bir işletim sistemi uyumlu adı (host name) bir de IRAF tanımlı (mantıksal) adı vardır: *mta*, *mtb* vb. Cihazlara IRAF'tan genellikle mantıksal adları ile ulaşılır. *tapecap* dosyasında kullanılan parametrelerle ilgili bilgiyi **page os\$zfiomt.c** komutu yardımı ile alabilirsiniz. IRAF sistem yöneticisi gerektiğinde bu dosyada ayarlamalar yapabilir. Aşağıdaki örnekleri inceleyiniz:

```
cl> devices (kullanabileceğiniz cihazları listeler)  
cl> allocate mta (mta tape sürücüsünü kullanıma tahsis eder)  
cl> rewind mta (mta sürücüsündeki tape cihazını başa sarar)  
cl> devstatus mta (mta cihazının o andaki durumu ile ilgili bilgi verir)  
cl> deallocate mta (mta sürücüsünü kullanım dışı bırakır)
```

3.2. Teyp Ortamına Yazmak ve Teyp Bilgi Yoğunluğunu Ayarlamak

Exabyte ve 9-izli cihazları da içeren bazı teyp cihazları birden fazla yoğunluk değerlerini destekler. Bu tür teyplerden okuma işlemi için değil ancak teybe yazım işlemi için teyp bilgi yoğunluğunu doğru bir şekilde belirtmek çok önemlidir.

Belirli bir yoğunluğa sahip yeni bir teyp cihazına bilgi yazma işlemi işletim sistemi bağımlıdır. Yoğunluk bilgisini işletim sistemine IRAF'ın içinden bildirebiliriz. Eğer böyle bir bilgi yazdırma komutunda yer almıyorsa, eski yoğunluk bilgisi geçerli olmaya devam edecek demektir. Aşağıdaki örnekleri inceleyelim:

```
cl> allocate mta (mta tape sürücüsünü tahsis eder)
cl> wfits mta6250 <image1> new+ (bir görüntüyü 6250 bpi yoğunluklu teybe yazar)
cl> wfits mta <image2> new- (ikinci bir görüntüyü teybe ekler)
```

3.3. FITS Formatlı Veriler Üzerinde Bazı Basit İşlemler

3.3.1. FITS Görüntülerini Okumak

IRAF, genel FITS formatını desteklemektedir. FITS formatındaki görüntüler IRAF'ta DATAIO paketinde yer alan RFITS taskı yardımı ile okunur. Bu taskın parametreleri aşağıda listelenmiştir:

<code>fits_file</code>	=	<code>'mta'</code>	FITS data source
<code>file_list</code>	=		File list
<code>iraf_file</code>	=	<code>' '</code>	IRAF filename
<code>(make_image</code>	=	<code>yes)</code>	Create an IRAF image?
<code>(long_header</code>	=	<code>no)</code>	Print FITS header cards?
<code>(short_header</code>	=	<code>yes)</code>	Print short header?
<code>(datatype</code>	=	<code>' '</code>	IRAF data type
<code>(blanc</code>	=	<code>0.</code>	Blank value
<code>(scale</code>	=	<code>yes)</code>	Scale the data?
<code>(oldirafname</code>	=	<code>no</code>	Use old IRAF name in place of iraf_file?
<code>(offset</code>	=	<code>0)</code>	Tape file offset
<code>(mode</code>	=	<code>'q1'</code>	

Aşağıda yer alan örnekte, FITS formatında data içeren bir teypten data okuma ve hard diske aktarma işlemleri gösterilmiştir (teypte 999'dan az sayıda dosya olduğu varsayılarak). Burada, çıktı görüntülerinin adı "n10001, n10002, n10003,....." olacaktır. Çıktının pixel tipi ise giriş datasının tipine (tamsayı, gerçel sayı, vb.) bağlı olacaktır. RFITS taskının çalışması biraz zaman alacağından, bu işin arka plana (*background job*) atılması ve üretilecek terminal çıktısının (kısa header listesi ve muhtemel hata durumunda hata mesajları) bir dosyaya yönlendirilmesi önerilir. Bu dosya daha sonra istendiğinde incelenebilir veya yazdırılabilir.

```
cl> show imdir(pixel dosyalarının bulunduğu dizini gösterir)
cl> diskspace(bu komut ile görüntüler için yeteri kadar yer
              olup olmadığını kontrol edilebilir)
cl> allocate mta(mta teyp sürücüsünü tahsis eder)
cl> rfits mta 1-999 n1 >& tlist &(teypteki tüm görüntüleri okur ve üretilen
                                kısa header listesini terminal yerine
                                isimli dosyaya aktarır. Bu işin arka
                                yapıldığını unutmayınız! Bunu sağlayan
                                "&")
```

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

parametresidir.)
cl> **page tlist** (tlist dosyasının içeriğini sayfa sayfa gösterir)

Teypte bulunan FITS dosyalardan birine ait header bilgisinin tamamını ekrana yazdırmak mümkündür. Aşağıda yer alan örnekte teypte bulunan 35. dosyanın header bilgisi listelenecek ve çıktı PAGE taskına yönlendirilecektir; bu şekilde çıktı kolayca incelenebilecektir.

cl> **rfits mta 35 make- l+ | page**

Burada *make-* (ya da *make_image=no*) parametresi sadece listenin üretilmesini sağlar, diske bir şey yazılmaz. Bu yöntemle elde edilen header bilgileri, aynı FITS dosya için IMHEADER taskı ile elde edilen bilgilerden farklı olacaktır.

FITS dosyalar teypten okunurken, datanın giriş formatı ne olursa olsun, çıkış formatı komut satırında belirtilebilir. Birçok kullanıcı, daha sonraki işlemlerde dataların tipi ile ilgili herhangi bir karışıklık olmaması için, tüm dataları IRAF'a gerçel (real) tipte aktarmayı daha kullanışlı bulurlar.

cl> **rfits mta 1-10,30-56 dt data=r > dtlist &**

Bu örnekte görüldüğü gibi çıktı datalarının tipi "real" olacaktır ve sadece seçilen dosyalardan (1-10,30-56) okuma yapılacaktır. Bu işlemin sonucunda üretilecek çıktı yine bir dosyaya (*dtlist*) yönlendirilmiştir ancak bu kez muhtemel hata mesajları dosyada değil ekranda yer alacaktır.

WFITS komutu, IRAFNAME anahtar kelimesini kullanarak görüntü dosyasının eski adını FITS header dosyasına yazar. Bu anahtar kelimenin değeri (yani dosya adı) *oldirafname=yes* tanımlaması yapıldıktan sonra RFITS ile okunabilir.

Şimdi, önceden teybe kaydedilmiş olan belirli bazı görüntüleri diske taşımak istediğimizi varsayalım. Teypte kayıtlı FITS görüntüleri ada göre değil dosya konumuna göre okunur, unutmayınız! Bu durumda önce teypte yer alan dosyaları isimlerine göre listeleyip, daha sonra okunacak dosyaları seçmeliyiz ve nihayet RFITS komutunu uygulamalıyız. Aşağıdaki örnekte bu yapılmıştır.

cl> **rfits mta 1-999 make- old+ | page**(teypte bulunan dosyaları IRAFNAME parametresindeki isimlere göre listeler)
cl> **rfits mta 20-25,45-47 junk old+**(teypten dosyaları okur ve diske yazarken görüntü adını IRAFNAME parametresindeki adla değiştirir)

RFITS ile sadece teypte değil diskte bulunan görüntü dosyaları da okunabilir. Örnekte görüldüğü gibi "fits001, fits002, fits003 ..." isimli FITS görüntüleri IRAF görüntülerine dönüştürülecektir.

cl> **rfits fits* 1 m31**

Çok uzun FITS header bilgileri durumunda, bazı kayıtlardaki parametrelerin değerleri kesilebilir ve IRAF görüntüsünün header bilgisinde yer almayabilir. *min_lenuserarea* çevre değişkeni kullanılarak görüntü header bilgisinin alanı büyütülebilir ve görüntüleri okuma işlemi yeniden yapılabilir. Bu değişkenin kabul edilmiş değeri 20000'dir (yaklaşık 250 kayıt).

```
cl> set min_lenuserarea=40000
```

3.3.2. FITS Görüntülerini Yazmak

Daha önceki bölümde yapılan işin tersini yapmak yani bir IRAF görüntüsünü bir FITS dosyası olarak teybe yazmak mümkündür. Bu işlem DATAIO paketinde bulunan WFITS taskı ile yapılır. Ancak bu noktada kullanıcıyı uarmakta fayda var. Datanızı teybe yazdıktan sonra bir kez daha ters işlemi yapınız yani teypten okuyup diskinize yazdırınız ve diskteki kopyası ile karşılaştırınız! Böylece datanızda bir bozulma olup olmadığından emin olursunuz.

WFITS taskının sistem tarafından tanımlanan parametreleri aşağıda listelenmiştir.

iraf_files	=	IRAF images
fits_files	= ' '	FITS filename
newtape	=	Blank tape?
bscale	= 1.	FITS bscale
bzero	= 0.	FITS bzero
(make_image	= yes)	Create a FITS image?
(long_header	= no)	Print FITS header cards?
(short_header	= yes)	Print short header?
(bitpix	= 0)	FITS bits Per pixel
(blocking_fac	= 0)	FITS tape blocking factor
(scale	= yes)	Scale data?
(autoscale	= yes)	Auto_scaling?

Dataların indirgenmesi işlemi bittikten sonra kullanıcının diskinde birçok görüntü seti oluşmuş olabilir. Bunların IEEE'nin kayan noktalı FITS formatında yeni bir Exabyte teybine aktarılması istenebilir. Bunun nasıl yapılabileceğini bir örnek üzerinde inceleyelim. Kullanılacak örnek görüntülerin pixel data tipinin gerçel olduğunu ve "bred0001, bred0002,....." ve "rred0001, rred0002,....." şeklinde isimlendirildiğini varsayalım. Önce WFITS taskının girdisi olacak ve görüntü isimlerinin listesini içeren bir text dosyası oluşturmalıyız. Bu dosya üzerinde gerektiğinde değişiklik (listeye yeni görüntü adı ekleme veya çıkarma) yapılabilir. WFITS taskının kısa header bilgilerinden oluşan çıktısı terminal yerine bir dosyaya yönlendirilecektir.

```
cl> files rred*.imh > inlist (girdi datasının listesini oluşturur)
cl> files bbred*.imh >> inlist (aynı listeye ekleme yapılır)
cl> allocate mtd (mtd isimli teyp sürücüyü tahsis eder)
cl> unlearn wfits (WFITS taskının kabul edilen değerlerini
                  kullanmayı sağlar)
cl> wfits @inlist mtd new+ > tlist & (datayı teybe yazar)
```

Datayı teybe eklemek için *new* parametresini, *new=no* veya *new-* şeklinde tanımlamak gerekir. Aşağıdaki örnekte, her görüntünün otomatik olarak scale edilmesini sağlamak için *bitpix=32* ve *blocking_factor=1* olarak tanımlanmıştır. İşlemin çıktısı varolan dosyaya yönlendirilmiştir.

```
cl> wfits xred1??.imh mta new- bitpix=32 block=1 >> tlist &
```

```
Warning: Default bitpix overridden.
Bitpix set to: 32
Warning: FITS tape blocking factor is 1
```

Burada, iki özel parametrenin – *bitpix* ve *blocking_factor* - değeri değiştirildi. Komutun icra edilmesinden sonra ekrana kullanıcıyı bilgilendiren bir uyarı mesajı gelir.

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

WFITS ile teybe istenilen bir dosya numarasından başlayarak data yazdırmak mümkündür. Örneğin 50 numaralı dosyaya yazmak için aşağıdaki işlemi uygularız (ancak bu işlemin Exabyte teyplerle çalışırken geçerli olmadığını belirtmek gerekir). Aksi belirtilmediği için ve dev\$pix IRAF test görüntüsünün önceden tanımlı data tipi tamsayı olduğundan, FITS dosyaya bilgiler 16 bitlik tamsayı formatında kaydedilecektir:

```
cl> wfits dev$pix mta[50]
```

Kullanıcı FITS görüntüleri sadece teybe yazmakla sınırlı değildir, bunları diske yazmak da mümkündür.

```
cl> wfits m31red,m31blue fits
```

Bu örnekte, *m31red* ve *m31blue* görüntüleri IEEE'nin kayan nokta formatında FITS dosyalar olarak diske yazdırılmıştır; dosyaların adı sırası ile *fits001* ve *fits002* olacaktır.

(NOT: diskteki dosyaların isimleri "mt" ile başlayamaz çünkü bu durumda IRAF onların diske değil teybe yazılacağını varsayar.)

3.4. Text (Metin) Dosyalar

Daha önceki bölümlerde text dosyalarla ilgili basit işlemlerden kısaca bahsedilmişti. Metin dosyaların okunması ve yazılması DATAIO paketindeki ilgili taskların kullanılması ile olur. Diğer bazı tasklar ise IRAF görüntülerini text dosyaya çevirme ve tersi işlemlerini yapmaktadır. Bu taskların kullanımı çok açık ve kolaydır. Aşağıda isimleri yer alan tasklarla ilgili daha fazla bilgi için on-line elkitabı sayfalarına bakılması önerilir.

Rcardimage	cardimage dosyasını text dosyaya dönüştürür
Rtextimage	text dosyayı IRAF görüntüsüne dönüştürür
Wcardimage	text dosyayı cardimage dosyaya dönüştürür
Wtextimage	IRAF görüntüsünü text dosyaya dönüştürür

3.5. Standart Olmayan Data Formatları

Kullanıcıların sıklıkla FITS veya text formatında olmayan ama IRAF görüntü formatına dönüştürülmesi gereken dataları vardır. PROTO paketindeki IRAFIL taskı bazı binary dataların IRAF görüntülerine dönüştürülmesi için kullanılır. Bu task 8 veya 16-bitlik pixel dataları ile çalışır. Bu dönüştürme sırasında header dosyasındaki bazı bilgiler silinebilir. Dönüştürülecek datanın önceden diske aktarılmış olması gerekir.

IRAFIL, MTEXAMINE ve REBLOCK (DATAIO paketindedir) birleşik kullanımı ile teypte bulunan binary data doğrudan IRAF görüntüsüne dönüştürülebilir.

MTEXAMINE	teypte bulunan dosyanın kayıt yapısını görüntüler
REBLOCK	teypteki dosyaları diske aktarır ve eğer gerekli
ise	<i>reblocking</i> veya <i>byte swapping</i> yapar
IRAFIL	diske aktarılan binary dosyaları IRAF görüntülerine dönüştürür.

Kullanıcı IMFORT arabirimini kullanarak kendi dönüştürme programlarını yazabilir veya mevcut SSP teyp okuma programlarına benzer programlar geliştirebilir.

3.6. TAR Dosya Formatı

Diskte veya teypte bulunan “tar” formatındaki verileri okumak ve yazmak için SOFTTOOLS paketindeki RTAR ve WTAR taskları kullanılır. Bunların uygulanması esnasında cihaz isimlerine dikkat edilmeli; IRAF’ın kısaltması olan “mta” gibi isimler değil, ana işletim sisteminin kullandığı cihaz isimleri kullanılmalıdır!

```
rtar          tar formatındaki arşiv dosyasını okur
wtar          tar formatındaki arşiv dosyasını yazar
```

IRAF’ta tar formatın üç farklı biçimi ile karşılaşabiliriz:

- IRAF ile çalışan çoğu harici yazılım paketi tar formatında gelir.
- IRAF’ın ürettiği veri tabanı dosyaları (ASCII) tar formatında kaydedilebilir ve böylece bir platformdan diğer bir platforma taşınabilir.
- Teypten okunan ve diske FITS formatlı dosyalar olarak kaydedilen IRAF görüntüleri de tar formatında bir dosyaya kaydedilebilir ve bu dosya daha sonra başka bir makineye taşınabilir. IRAF görüntülerini doğrudan tar formatına dönüştürmek tavsiye edilmez!

3.7. Bazı Ek G/Ç (I/O) Taskları

Bu bölümde DATAIO paketinde yer alan MTEXAMINE, T2D ve REBLOCK tasklarından bahsedilecektir. MTEXAMINE, teyp ve disk dosyalarını incelemek için kullanılır.

T2D taskı ile teyp dosyaları hızlıca diske kopyalanabilir. Bu task özellikle problemlili teyplerle çalışırken işe yarar. Bozuk teypte bulunan datayı diske aktardıktan sonra kullanılacak bölümünü açmak daha kolay olur.

REBLOCK taskı birçok işe yarar. Datayı diske okunduğu gibi swap etmek veya datayı yeniden bloklamak mümkündür. Bu task ile ayrıca teypten teybe data kopyalanabilir (gerekli ise yoğunluk değiştirilebilir).

4. IRAF’ta Grafik

IRAF’ın genel amaçlı çizim taskları PLOT paketindedir ancak grafik üretimi sadece bunlarla sınırlı değildir.

Kullanılacak terminalin tipini IRAF’a tanıtmak için STTY taskındaki tanımlamalar gerektiği gibi yapılmış olmalıdır. Bunu ya IRAF’ı başlattığımızda ya da IRAF’ta çalışırken grafik ile ilgili bir uygulamaya başlamadan önce yapabilirsiniz.

Üzerinde çalıştığımız grafik, grafik penceresi için ayrılmış olan bellek alanında (*frame buffer*) yer alır. Bu alanın büyüklüğü **cmbuflen** çevre değişkeni tarafından tanımlanır. Eğer grafik için ayrılan alan yeterli değilse kapsamlı grafikler eksik görüntülenebilir; bellek alanını büyütürük bu problem çözülebilir. Takip eden örnek bunun nasıl yapılacağını göstermektedir. Buna benzer yeni değerlerin tanımlanmasından sonra bu değerlerin geçerli olması için, örnekte de gösterildiği gibi “**gflush**” komutu kullanılır.

```
cl> show cmbuflen
cl> reset cmbuflen = 256000
```

cl> **gflush**

4.1. Etkileşimli Grafik Cursor Modu (Interactive Graphics Cursor Mode)

IRAF'ta birçok taskın ürettiği grafik önceden tanımlı tuşlar ve grafik penceresindeki cursor ile etkileşimli olarak incelenebilir ve işlenebilir. Bir kısım grafik taskı cursor'ı doğrudan grafik penceresine konumlandırırken başka bir kısım task ise sadece grafiği çizer, cursor yine text penceresindedir. Bu ikinci grup tasktan da istenirse “=**gcur**” komutu ile etkileşimli cursor moduna geçilebilir.

Bir grafik arabirimi olan etkileşimli cursor modu bir temel cursor komutları (*cursor keystrokes*) setine sahiptir. Cursor komutları ya tek karakterlik büyük harflerden ya da “:.” işaretleri ile birlikte kullanılan komutlardan oluşur (Şekil 4).

Etkileşimli cursor modundayken tüm cursor komutlarını listelemek için **:.help** komutu kullanılır. Belirli bir task için geçerli cursor komutları ise **'?** komutu ile listelenebilir.

cl> **phelp cursors**(on-line yardım sayfalarından hem grafiklerle hem de cursor komutları ile ilgili bilgi getirir)

Şimdi aşağıdaki uygulamaları inceleyiniz:

```
cl> stty           (terminal tipini kontrol eder)
cl> stty stermjh  (terminal tipini stermjh olarak tanımlar)
cl> contour dev$pix (dev$pix görüntüsü ile ilgili grafik çizer)
cl> =gcur         (etkileşimli grafik moduna aktif hale getirir)
:.help          (tüm cursor komutlarını listeler)
...             (global cursor komutları ile deneyiniz)
0 ???          (grafiğin tamamını tekrar listeler)
:.snap <yazıcı adı> (grafiğin çıktısını belirtilen yazıcıya yollar)
=              (grafiğin çıktısını önceden tanımlı olan yazıcıya yollar)
q             (herhangibir küçük harf ile çıkılır)
```

4.2. Tek Boyutlu Grafik Taskları

Aşağıdaki listeden de görüldüğü gibi PLOT paketinde grafik üreten pek çok task vardır. Bunlardan sadece IMPLICIT etkileşimlidir. Diğer tasklardan etkileşimli grafik modunu aktif duruma getirmek için “=**gcur**” komutu kullanılır.

```
graph          bir veya birden fazla görüntü bölümünün grafiğini üretir
implot         görüntülerin satır ve sütun grafiklerini çizer
pcol           bir görüntünün tek bir sütununun grafiğini çizer
pcols          bir sütun aralığının ortalamasının grafiğini çizer
phistogram     bir ya da birden fazla görüntünün histogramını çizer ya da yazıcıya gönderir
pradprof       bir yıldızın radyal profilini çizer veya listeler
prow           bir görüntünün satır grafiğini çizer
prows         bir satır aralığının ortalamasının grafiğini çizer
pvector        bir 2D görüntüsünün herhangi bir vektörünün grafiğini çizer
```

Veri indirgemede iki boyutlu datayı incelemek için kullanılan en önemli task IMPLICIT'tur. Bu task hem kendi cursor komut setini (küçük harfler ve “:.” karakteri) hem de her task için geçerli etkileşimli cursor komut setini kullanır. Aşağıdaki örneği inceleyelim:

```
cl> implot dev$pix(M51 görüntüsünün merkez çizgisinin grafiğini çizer)
?          (yardım - cursor komutlarını listeler)
q        (cursor yardım penceresinden çıkar)
...       (cursor komutları ile deneyiniz)
q        (tasktan çıkar)
```

GRAPH taskı ile ASCII formatındaki dosya veya görüntülerin grafiği elde edilebilir. GRAPH etkileşimli değildir ancak kullanıcıya eksenleri kontrol etme imkanı verir, böylece grafik yayına hazır şekilde üretilebilir.

A	draw and label the axes of current viewport
B	backup over last instruction in frame buffer
C	print the cursor position
D	draw a line by marking the endpoints
E	expand plot by setting window corners
F	set fast cursor (for HJKL)
H	step cursor left
J	step cursor down
K	step cursor up
L	step cursor right
M	move point under cursor to center of screen
P	zoom out (restore previous expansion)
R	redraw the screen
T	draw a text string
U	undo last frame buffer edit
V	set slow cursor (for HJKL)
W	select WCS at current position of cursor
X	zoom in, X only
Y	zoom in, Y only
Z	zoom in, both X and Y
<	set lower limit of plot to the cursor y value
>	set upper limit of plot to the cursor y value
\	escape next character
:	set cursor mode options
:!	send a command to the host system
=	short for "!.snap"
0	reset and redraw
1-9	roam

Cursor Mode Commands

::axes[+-]	draw axes of viewport whenever screen is redrawn
::case[+-]	enable case sensitivity for keystrokes
::clear	clear alpha memory (e.g, this text)
::cursor n	select cursor
::gflush	flush plotter output
::help	print help text for cursor mode
::init	initialize the graphics system
::markcur[+-]	mark cursor position after each cursor read
::off [keys]	disable selected cursor mode keys
::on [keys]	enable selected cursor mode keys
::page[+-]	enable screen clear before printing help text
::read file	fill frame buffer from a file
::show	print cursor mode and graphics kernel status
::snap [device]	make hardcopy of graphics display
::txqual qual	set character generator quality (normal,l,m,h)
::txset format	set text drawing parameters (size,up,hj,vj,etc)
::xres=value	set X resolution (stdgraph only)
::yres=value	set Y resolution (stdgraph only)
::viewport x1 x2 y1 y2	set workstation viewport in world coordinates
::write[!][+]file	save frame buffer in a spool file
::zero	reset viewport and redraw frame

Şekil 4:Etkileşimli Grafik Cursor Komutları (Not: orijinal olarak verilmiştir)

4.3. İki Boyutlu Grafik Taskları

IRAF'taki birçok iki boyutlu grafik taskını etkileşimli kullanabilmek için '=gcur' komutunu uygulamak gerekir .

```
contour    bir görüntünün çizgi grafiğini üretir
hafton     Bir görüntünün half-tone grafiklerini üretir
surface    Bir görüntünün yüzey grafiğini üretir
velvect    Bir hız alanının temsili grafiğini üretir
```

CONTOUR taskı ile ilgili aşağıdaki örneği uygulayınız ve inceleyiniz ('\ karakterinin kullanımını daha önceki bölümlerden hatırlayacaksınız).

```
cl> display dev$pix 1(M51'i görüntüler)
cl> contour dev$pix \
xres=256 yres=256 perim- fill+ label- ceil=500 dev=imdg
```

4.4. Grafik Metakod Dosyaları ile İlgili Tasklar

Grafik çıktısı bir dosyaya kaydedilebilir. Bu, 'metakod' dosya olarak isimlendirilir ve iki yoldan yaratılabilir:

- Grafik task yardımı ile yaratıldıktan sonra ">G" komutu ile bir dosyaya (metakod) yönlendirilir
- Üretilen grafik, etkileşimli cursor modundaki "::.write" komutu ile doğrudan metakod dosyasına aktarılır

```
cl> contour dev$pix >G meta (grafik meta isimli dosyaya yönlendirilir)
cl> surface dev$pix         (bir grafik üretilir)
cl> =gcur                   (etkileşimli cursor moduna geçilir)
: .write meta               (üretilen grafiği meta isimli dosyaya ekler)
: .read meta                (meta dosyasındaki grafikleri görüntüler)
q                            (tasktan çıkılır)
```

Aşağıdaki tasklar ile metakod dosyaların içine bakılır veya bunların içindeki grafikler görüntülenebilir:

```
gkidir      metakod dosyasının içeriğini listeler
gkiextract  metakod dosyanın içindeki frame'leri tek tek çıkarıp alır
gkimosaic   metakod dosyada bulunan frame'leri bir sayfaya sığacak
            şekilde sıkıştırır
stdgraph    metakod dosyayı standart grafik cihazında gönderir
stdplot     metakod dosyayı standart çizim cihazına (plotter)
gönderir
```

Kullandığımız IRAF sisteminde bir örnek metakod dosya mevcuttur. Aşağıdaki örneklerde bu dosya kullanılmıştır.

```
cl> gkidir dev$vdm.gki (dev$vdm.gki metakod dosyasının içeriğini listeler)
cl> gkimos dev$vdm.gki (metakod dosyasının içeriğini sıkıştırılmış formda
                        terminalde görüntüler)
cl> gkiextract dev$vdm.gki 2 | stdgraph
                        (metakod dosyasındaki 2. grafiği görüntüler)
cl> stdgraph dev$vdm.gki(tüm dosyaları terminalde görüntüler)
```

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

```
cl> stdplot dev$vdm.gki(tüm dosyaları önceden tanımlı yazıcıya gönderir)
```

5.Görüntüler ile İlgili İşlemler ve Komutlar

IMAGES paketinin bir altpaketi olan TV, görüntülerle ilgili komutları içerir. Workstation ortamında çalışan bir kullanıcı, görüntüyü önce frame tampon belleğine aktarır ve daha sonra XIMTOOL veya SAOimage görüntüleme yazılım aracının komutlarını kullanarak görüntü üzerinde çeşitli işlemler yapar. UNIX ortamında “**man imtool**” komutu ile XIMTOOL’un kullanımını konusunda bilgi alınabilir.

Görüntüleme işlemlerine geçmeden önce **stdimage** çevre değişkeninin değeri, frame tampon bellek büyüklüğü ile uyumlu olmalıdır veya doğru görüntüleme cihazına yönlendirilmiş olmalıdır. Bunları yapmak gerekirse GDEVICES taskı kullanılabilir.

```
cl> show stdimage(geçerli değeri gösterir)
```

```
cl> set stdimage=imt800(stdimage çevre değişkeninin değerini tanımlar)
```

```
cl> set stdimage=iism70v(görüntüleme cihazını tanımlar)
```

5.1. Görüntüler ile İlgili İşlemler

IRAF’ta görüntüler DISPLAY taskı ile görüntülenir. Bu task ile XIMTOOL’da en fazla dört frame, SAOimage’te ise ancak bir frame için tampon bellek tahsis edilmiştir.

```
cl> display dev$pix 1 (görüntü 1. frame’e aktarılır)
```

Frame’e aktarılan görüntü tanım gereği frame’in tam ortasında yer alır. Eğer görüntü frame tampon bellek alanından daha büyükse, görüntünün sadece merkez bölgesi frame’de yer alır. Tersisi durumunda ise, görüntü çerçeve içine alınarak frame’e yerleştirilir.

Aşağıdaki örnekler büyük ve küçük görüntü durumlarında, görüntülerin bozulmadan frame’e aktarılması için yapılabilecek işlemler içermektedir.

```
cl> set stdimage=imt256 (görüntü büyüklüğünden daha küçük  
bir alan tanımlar)
```

```
cl> display dev$pix 1 fill+
```

```
cl> set stdimage=imt1024(görüntü büyüklüğünden daha büyük  
bir alan tanımlar)
```

```
cl> display dev$pix 1 fill+ order=0
```

DISPLAY taskının birçok seçeneği vardır. Daha fazla bilgi için taskın parametreleri ve on-line yardım sayfaları incelenebilir.

5.2. Görüntü Ortamında Etkileşimli Cursor Kullanımı

Etkileşimli cursor komutlarına sahip tasklardan görüntü ortamına (XIMTOOL veya SAOimage) müdahale etmek mümkündür. Cursor’ın etkileşimli kullanımı **stdimcur** çevre değişkeni ile kontrol edilmektedir. **stdimcur**’un normal değeri “**stdimage**” dir; bunun anlamı etkileşimli cursor’ın görüntü ortamında (XIMTOOL veya SAOimage) geçerli olmasıdır. Bunun yanı sıra **stdimcur**, “**text**” veya “**stdgraph**” olarak da tanımlanabilir. Bu esneklik kullanıcıya, görüntü ve grafik ortamlarında aynı anda çalışma imkanını verir. Aşağıdaki örnekleri inceleyelim:

```
cl> show stdimcur (geçerli olan değeri gösterir)
cl> set stdimcur=stdimage(görüntü modunu aktif yapar)
cl> set stdimcur=stdgraph(grafik ortamında cursor kullanımını aktif hale getirir)
cl> set stdimcur=text (text modunu aktif yapar)
```

Eğer kullanılan görüntüleme aracı birden fazla görüntü frame'ini destekliyorsa, etkileşimli cursor modunda kullanılacak görüntü "kontrol" karakterleri şunlardır:

- Ctrl-F: bir sonraki frame'e geçilir
- Ctrl-R: bir önceki frame'e dönülür
- Ctrl-B: belirli frame'leri arka arkaya görüntüler (*blink*)

5.2.1. Görüntü Ortamında Cursor Konumlarının Belirlenmesi

LISTS paketindeki RIMCURSOR taskı ile o anda tampon bellekte bulunan görüntüde cursor'ın koordinatları belirlenebilir. Ayrıca, belirlenen koordinat listesi bir başka taska giriş bilgisi olarak aktarılabilir. Taskın çıktısı bir dosyaya da yönlendirilebilir. RIMCURSOR taskından Ctrl-Z veya Ctrl-D tuşları ile çıkılır. Aşağıdaki örnekleri inceleyiniz:

```
cl> display dev$pix 1 (görüntü XIMTOOL penceresine aktarılır)
cl> rimcursor (koordinatlar okunur (belleğe aktarılır))
cl> rimcursor > coordlist (koordinatlar bir dosyaya yönlendirilir)
```

Son komutla üretilen ve bir dosyaya aktarılan koordinatlar aşağıda listelenmiştir. Bunların içinde en çok kullanılan ilk iki değerdir.

- x ve y pixel koordinatları
- frame numarası (1,2 vb.)
- okuma işlemini kesen komut

5.2.2. Görüntüleri İncelemek ve Düzeltmek

TV paketinde, görüntüleri etkileşimli incelemek ve düzeltmek için bir çok kullanışlı task vardır. Bu kullanım kılavuzunda bunlarla ilgili çok fazla bilgi olmamasına rağmen kullanıcının on-line yardım sayfalarına başvurması ve bunlarla ilgili çeşitli örnekler yapması önerilir.

```
IMEXAMINE frame'de yer alan görüntüyü bir çok yoldan inceleme imkanı verir
IMEDIT frame'de yer alan görüntü üzerinde çeşitli düzeltmeler yapılmasını sağlar
TVMARK frame'de yer alan görüntü üzerinde çizim ve koordinat listesi üretme ve ekleme için kullanılır
```

Aşağıdaki örnekte, IMEXAMINE taskının birkaç etkileşimli cursor komutu ile ilgili uygulama bulacaksınız. Bu task ile, etkileşimli cursor komutlarının yanı sıra etkileşimli grafik cursor komutları da kullanılabilir !

```
cl> imex dev$pix (etkileşimli görüntü cursor modunu aktif duruma getirir)
? (cursor komutlarını listeler)
```

z	(cursor'ın altındaki bölgenin pixel değerlerini görüntüler)
a	(cursor'ın altındaki yıldız ile ilgili bilgi verir)
s	(bir bölgenin yüzey grafiğini çizer)
: epar	(s komutu ile ilgili parametre listesini düzeltme imkanı verir)
l	(satır grafiği (<i>line plot</i>) üretir)
: nave 20	(20 satırın ortalamasını alır)
g	(grafik cursor moduna geçilir)
Z	(grafiğe zoom yapar)
o	(ilk (orijinal) grafiğe döner)
i	(etkileşimli görüntü cursor moduna geri döner)
r	(incelenen cismin radyal profilini çizer)
q	(tasktan çıkar)

(NOT: “Görüntü Ortamında Etkileşimli Cursor Kullanımı” bölümü veri indirgeme ve analizinde çok önemli bir yer tutar. Daha detaylı bilgi için yazarların hazırlanmakta olan “IRAF’ta Görüntüler Üzerinde İşlemler” isimli elkitabına başvurabilirsiniz.)

Ek A IRAF Paketleri (Sürüm 2.10)

clpackage.clpackage:

dataio	Data format conversion package (RFITS, etc.)
dbms	Database management package (not yet implemented)
images	General image processing package
language	The command language itself
lists	List processing package
local	The template local package
obsolete	Obsolete tasks
noao	The NOAO optical astronomy packages
plot	Plot package
proto	Prototype or interim tasks
softtools	Software tools package
system	System utilities package
utilities	Miscellaneous utilities package

clpackage.dataio:

bintxt	Convert a binary file to an IRAF text file
mtxamine	Examine the structure of a magnetic tape
rcardimage	Convert a cardimage file into a text file
reblock	Copy a binary file, optionally reblocking
rfits	Convert a FITS image into an IRAF image
rtextimage	Convert text files to IRAF images
t2d	Fast tape to disk copy
txtbin	Convert an IRAF text file to a binary file
wcardimage	Convert text files to cardimage files
wfits	Convert an IRAF image into a FITS image
wtextimage	Convert an IRAF image to a text file

clpackage.images:

blkavg	Block average or sum a list of ND images
blkrep	Block replicate a list of images
boxcar	Boxcar smooth a list of 1 or 2D images
chpixtype	Change the pixel type of a list of images
convolve	Convolve a list of 1 or 2D images with a rectangular filter
fit1d	Fit a function to image lines or columns

fmedian	Quantize and median filter a 2D image or a list of
images	
fmode	Quantize and modal filter a 2D image or list of images
gauss	Convolve a list of 1 or 2D images with an elliptical
Gaussian	
geomap	Calculate a coordinate transformation
geotran	Geometrically transform a set of 2D images
gradient	Convolve a list of 1 or 2D images with a gradient
operator	
hedit	Header editor
hselect	Select a subset of images satisfying a boolean
expression	
imarith	Simple image arithmetic
imcombine	Combine images pixelbypixel using various algorithms
imcopy	Copy an image
imdebug	Image debugging package (currently undocumented)
imdelete	Delete a list of images
imdivide	Image division with zero checking and rescaling
imgets	Return the value of an image parameter as a string
imheader	Print an image header
imhistogram	Compute image histogram
imlintran	Linearly transform a list of 2D images
imrename	Rename one or more images
imshift	Shift a list of 2D images
imslice	Slice images into images of lower dimension
imstack	Stack images into a single image of higher dimension
imstatistics	Compute and print statistics for a list of images
imsum	Compute the sum, average, or median of a set of images
imsurfit	Fit a surface to a 2D image
imtranspose	Transpose a 2D image
laplace	Laplacian filter a list of 1 or 2D images
lineclean	Replace deviant pixels in image lines
listpixels	Convert an image section into a list of pixels
magnify	Magnify a list of 1D or 2D images
median	Median filter a 2D image or list of images
minmax	Compute the minimum and maximum pixel values in an
image	
mode	Modal filter a 2D image or list of images
register	Register a set of images
rotate	Rotate and shift a list of 2D images
sections	Expand an image template on the standard output
shiftnlines	Shift image lines
tv	Image display load and control package

clpackage.images.tv:

display	Load an image or image section into the display
iis	IIS image display control package
imedit	Examine and edit pixels in images
imexamine	Examine images using image display, graphics, and text
tvmark	Mark objects on the image display
wcslab	Overlay a displayed image with a world coordinate grid

clpackage.images.tv.iis:

blink	Blink two frames
cv	Control image device, display "snapshot"
cvl	Load image display (newer version of 'display')
erase	Erase an image frame
frame	Select the frame to be displayed
lumatch	Match the lookup tables of two frames
monochrome	Select monochrome enhancement
pseudocolor	Select pseudocolor enhancement
rgb	Select true color mode (red, green, and blue frames)

window Adjust the contrast and dc offset of the current frame
zoom Zoom in on the image (change magnification)

clpackage.language:

intro A brief introduction to IRAF

Language components:

break * Break out of a loop
case * One setting of a switch
commands A discussion of the syntax of IRAF commands
cursors Graphics and image display cursors
declarations Parameter/variable declarations
default * The default clause of a switch
else * Else clause of IF statement
for * Cstyle for loop construct
if * If statement
goto * Goto statement
logging Discussion of CL logging
next * Start next iteration of a loop
parameters Discussion of parameter attributes
procedure * Start a procedure script
return * Return from script with an optional value
switch * Multiway branch construct
while * While loop

Builtin Commands and Functions:

access Test if a file exists
back Return to the previous directory (after a chdir)
beep Send a beep to the terminal
bye Exit a task or package
cache Cache parameter files, or print the current cache
list
cd Change directory
chdir Change directory
cl Execute commands from the standard input
clbye A cl followed by a bye (used to save file)
descriptors)
clear Clear the terminal screen
defpac Test if a package is defined
defpar Test if a parameter is defined
deftask Test if a task is defined
dparam Dump a pset as a series of task.param=value
assignments
edit Edit a text file
ehistory Edit history file to reexecute commands
envget Get the string value of an environment variable
eparam Edit parameters of a task
error Print error code and message and abort
flprcache Flush the process cache
fprint * Print a line into a parameter
fscan * Scan a list
gflush Flush any buffered graphics output
hidetask Make a task invisible to the user
history Display commands previously executed
jobs Display status of background jobs
keep Make recent set, task, etc. declarations permanent
kill Kill a background job
logout Log out of the CL
lparam List the parameters of a task
mathfcns Mathematical routines
mktemp Make a temporary (unique) file name

osfn		Return the host system equivalent of an IRAF filename
package		Define a new package, or print the current package
names		
prcache		Show process cache, or lock a process into the cache
print		Format and print a line on the standard output
putlog		Put a message to the logfile
radix		Encode a number in the specified radix
redefine		Redefine a task
reset		Reset the value of an environment variable
scan	*	Scan the standard input
service		Service a query from a background job
set		Set an environment variable
show		Show an environment variable
sleep		Hibernate for a specified time
strings		String manipulation routines
stty		Set/show terminal characteristics
task		Define a new task
time		Print the current time
unlearn		Restore the default parameters for a task or package
update		Update a task's parameters (flush to disk)
wait		Wait for all background jobs to complete

clpackage.lists:

average	Compute the mean and standard deviation of a list
columns	Convert multicolumn file to separate files
lintran	Perform linear transformation of a list
rgcursor	Read the graphics cursor (makes a list)
rimcursor	Read the image display cursor (makes a list)
table	Format a list of words into a table
tokens	Break a file up into a stream of tokens
unique	Delete redundant elements from a list
words	Break a file up into a stream of words

clpackage.obsolete:

imtitle	Change the title of an image (noao.proto V2.9)
mkhistogram	List or plot the histogram of a data stream
(noao.proto V2.9)	
oimcombine	Combine images (images.imcombine V2.9)
radplot	PLot the radial profile of an object (noao.proto V2.9)

The previous package and release is listed in parenthesis

clpackage.plot:

calcomp	Plot metacode on a Calcomp pen plotter
contour	Make a contour plot of an image
crtpict	Generate greyscale plots of IRAF images
gdevices	List available imaging or other graphics devices
gkidecode	Decode metacode on the standard output
gkidir	Directory listing of metacode file
gkiextract	Extract individual frames from metacode file
gkimosaic	Condense metacode frames to fit on one page
graph	Graph one or more image sections or lists
hafton	Generate halftone plots of an image
imdkernel	Image display device (IMD) graphics kernel
implot	Plot lines and columns of images using cursors
nsppkernel	Plot metacode on a NSPP (NCAR) plotter device
pcol	Plot a column of an image
pcols	Plot the average of a range of image columns
phistogram	Plot or print the histogram of an image or list
pradprof	Plot or list the radial profile of a stellar object

prow	Plot a line (row) of an image
prows	Plot the average of a range of image lines
pvector	Plot an arbitrary vector in a 2D image
sgidecode	Decode an SGI format metacode file
sgikern	Simple graphics interface (SGI) graphics kernel
showcap	Show and decode graphcap entries
stdgraph	Plot metacode on the standard graphics device
stdplot	Plot metacode on the standard plotter device
surface	Make a surface plot of an image
velvect	Plot representation of a velocity field

clpackage.proto:

binfil	Create a binary file from an IRAF image
bscale	Linearly transform the intensities of a list of images
epix	Edit pixels in an image
fields	Extract specified fields from a list
fixpix	Fix bad pixels by linear interpolation from nearby pixels
hfix	Fix image headers with a user specified command
imalign	Register and shift a list of images
imcentroid	Compute relative shifts for a list of images
imcntr	Locate the center of a stellar image
imfunction	Apply a function to the image pixel values
imreplace	Replace pixels in a range by a constant
imscale	Scale an image to a specified (windowed) mean
inter	Interpolate for a value in a table of X,Y pairs
irafil	Create an IRAF image from a binary data file
joinlines	Join text files line by line
suntoiraf	Convert Sun rasters into IRAF images
wcsedit	Edit the image coordinate system
wcsreset	Reset the image coordinate system

clpackage.softools:

generic	Preprocess a generic source file
hdbexamine	Examine a help database
lroff	Lroff (lineroff) text formatter
mkhelpdb	Make (compile) a help database
mkmanpage	Make a manual page
mkpkg	Make or update an object library or package
mktags	Tag all procedure declarations in a set of files
mkttydata	Build cache for termcap/graphcap device entries
rmbin	Find/delete binary files in subdirectories
rmfiles	Find/delete files in subdirectories
rtar	Read a TAR format archive file
wtar	Write a TAR format archive file
xc	Compile and/or link a program
xyacc	Build an SPP language parser

clpackage.system:

allocate	Allocate a device, i.e., magtape drive mta, mtb, ...
concatenate	Concatenate a list of files
copy	Copy a file or files (use IMCOPY for imagefiles)
count	Count the number of lines, words, characters in a text file
deallocate	Deallocate a previously allocated device
delete	Delete a file or files (use IMDELETE to delete imagefiles)
devices	Print information on the locally available devices
devstatus	Print the status of a device (mta, mtb, ...)
directory	List the files in a directory

diskspace	Show how much disk space is available
files	Expand a file template into a list of files
gripes	Send suggestions, complaints, etc. to the system
head	Print the first few lines of a text file
help	Print online documentation
lprint	Print a file on the line printer device
match	Print all lines in a file that match a pattern
mkdir	Create a new directory
mkscript	Make a command script
movefiles	Move files to a directory
netstatus	Print the status of the local network
news	Page through the system news file
page	Page through a file
pathnames	Expand a file template into a list of OS pathnames
phelp	Paged HELP: collects and pages the output of HELP
protect	Protect a file from deletion
references	Find all help database references for a given topic
rename	Rename a file
rewind	Rewind a device (magtape)
sort	Sort a text file
spy	Show processor status
tail	Print the last few lines of a file
tee	Tee the standard output into a file
type	Type a text file on the standard output
unprotect	Remove delete protection from a file

clpackage.utilities:

curfit	Fit data with Chebyshev, Legendre or spline curve
detab	Replace tabs with tabs and blanks
entab	Replace blanks with tabs and blanks
lcase	Convert a file to lower case
polyfit	Fit polynomial to list of X,Y data
split	Split a large file into smaller segments
translit	Replace or delete specified characters in a file
ucase	Convert a file to upper case
urand	Uniform random number generator

Note: Language package keywords are starred. To get help on a keyword enclose it in quotes.

Ek B **NOAO Paketleri (Sürüm 2.10)**

(This list does not reflect any tasks added by the various patches to V2.10.)

noao:

artdata	Artificial data generation package [up]
astrometry	Astrometry package
astutil	Astronomical utilities package [up]
digiphot	Digital stellar photometry package [up]
focas	Faint object classification and analysis package
imred	Image reductions package [up]
mtlocal	Magtape i/o for special NOAO format tapes [up]
nobsolete	Obsolete tasks to be phased out in a future release
[up]	
nproto	Prototype (temporary, contributed) tasks [up]
observatory	Examine and define observatory parameters [up]
onedspec	One dimensional spectral red & analysis package [up]

rv Radial velocity analysis package [up]
surfphot Galaxy isophotal analysis package
twospec Two dimensional spectral red & analysis package [up]

noao.artdata:

gallist Make an artificial galaxies list
mk1dspec Make/add artificial 1D spectra
mk2dspec Make/add artificial 2D spectra using 1D spectra
templates
mkechelle Make artificial 1D and 2D echelle spectra
mkexamples Make artificial data examples
mkheader Append/replace header parameters
mknoise Make/add noise and cosmic rays to 1D/2D images
mkobjects Make/add artificial stars and galaxies to 2D images
mkpattern Make/add patterns to images
starlist Make an artificial star list

noao.astutil:

airmass Compute the airmass at a given elevation above the
horizon
asttimes Compute UT, Julian day, epoch, and siderial time
ccdtime Compute time required to observe star of given
magnitude
galactic Convert ra, dec to galactic coordinates
gratings Compute and print grating parameters
pdm Find periods in light curves by Phase Dispersion
Minimization
precess Precess a list of astronomical coordinates
rvcorrect Compute radial velocity corrections
setairmass Compute effective airmass and middle UT for an exposure
setjd Compute and set Julian dates in images

noao.digiphot:

apphot Aperture Photometry Package
daophot Dao CrowdedField Photometry Package
photcal Photometric Calibration Package
ptools Photometry Tools Package

noao.digiphot.apphot:

aptest Run basic tests on the apphot package tasks
center Compute accurate centers for a list of objects
centerpars Edit the centering parameters
daofind Find stars in an image using the DAO algorithm
datapars Edit the data dependent parameters
fitspf Model the stellar psf with an analytic function
fitsky Compute sky values in a list of annular or circular
regions
fitskypars Edit the sky fitting parameters
phot Measure magnitudes for a list of stars
photpars Edit the photometry parameters
polymark Create polygon lists for polyphot
polyphot Measure magnitudes inside a list of polygonal regions
polypars Edit the polyphot parameters
qphot Measure quick magnitudes for a list of stars
radprof Compute the stellar radial profile of a list of stars
wphot Measure magnitudes for a list of stars with weighting
lintran Linearly transform a coordinate list
pexamine Interactively examine or edit an apphot output file
txdump Dump select fields from an apphot output file

noao.digiphot.daophot:

addstar	Add artificial stars to an image using the computed psf
allstar	Group and fit psf to multiple stars simultaneously
centerpars	Edit the centering algorithm parameters
daofind	Find stars in an image using the DAO algorithm
daopars	Edit the daophot algorithms parameter set
daotest	Run basic tests on the daophot package tasks
datapars	Edit the data dependent parameters
fitskypars	Edit the sky fitting algorithm parameters
group	Group stars based on positional overlap and
signal/noise	
nstar	Fit the psf to groups of stars simultaneously
peak	Fit the psf to single stars
phot	Compute sky values and initial magnitudes for a list of
stars	
photpars	Edit the photometry parameters
psf	Fit the point spread function
seepsf	Compute an image of the point spread function
substar	Subtract the fitted stars from the original image
pappend	Concatenate a list of daophot databases
pconvert	Convert a text database to a tables database
pdump	Print selected fields from a list of daophot databases
grpselect	Select groups of a specified size from a daophot
database	
pexamine	Interactively examine and edit a daophot database
prenumber	Renumber stars in a daophot database
pselect	Select records from a daophot database
psort	Sort a daophot database

noao.digiphot.ptools:

istable	Is a file a table or text database file ?
pappend	Concatenate a list of apphot/daophot databases
pconvert	Convert from an apphot/daophot text to tables database
pdump	Print selected columns of a list of daophot/apphot
databases	
prenumber	Renumber a list of apphot/daophot databases
pexamine	Interactively examine and edit an apphot/daophot
database	
pselect	Select records from a list of apphot/daophot databases
psort	Sort a list of apphot/daophot databases
pttest	Run basic tests on the ptools package tasks
tbappend	Concatenate a list of apphot/daophot tables databases
tbdump	Print selected columns of a list of tables databases
tbrenumber	Renumber a list of apphot/daophot tables databases
tbselect	Select records from a list of apphot/daophot tables
databases	
tbsort	Sort a list of apphot/daophot tables databases
txappend	Concatenate a list of apphot/daophot text databases
txdump	Print selected columns of a list of apphot/daophot text
databases	
txrenumber	Renumber a list of apphot/daophot text databases
txselect	Select records from a list of apphot/daophot text
databases	
txsort	Sort a list of apphot/daophot text databases

noao.imred:

argus	CTIO ARGUS reduction package
bias	General bias subtraction tools
ccdred	Generic CCD reductions
ctioslit	CTIO spectrophotometric reduction package
dto	Density to Intensity reductions for photographic plates

echelle	Echelle spectra reductions (slit and FOE)
generic	Generic image reductions tools
hydra	KPNO HYDRA (and NESSIE) reduction package
iids	KPNO IIDS spectral reductions
irred	KPNO IR camera reductions
irs	KPNO IRS spectral reductions
kpnocoude	KPNO coude reduction package (slit and 3 fiber)
kpnoslit	KPNO low/moderate dispersion slits (Goldcam,
RCspec,Whitecam)	
specred	Generic slit and fiber spectral reduction package
vtel	Solar vacuum telescope image reductions

noao.imred.argus:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
aprecenter	Recenter apertures
apresize	Resize apertures
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plots of spectra
continuum	Fit the continuum in spectra
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
identify	Identify features in spectrum for dispersion solution
msresp1d	Create 1D response spectra from flat field and sky
spectra	
refspectra	Assign wavelength reference spectra to other spectra
reidentify	Automatically identify features in spectra
sapertures	Set or change aperture header information
sarith	Spectrum arithmetic
scombine	Combine spectra having different wavelength ranges
scopy	Select and copy apertures in different spectral formats
setairmass	Compute effective airmass and middle UT for an exposure
setjd	Compute and set Julian dates in images
slist	List spectrum header parameters
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
doargus	Process ARGUS spectra
demos	Demonstrations and tests

noao.imred.bias:

colbias	Fit and subtract an average column bias
linebias	Fit and subtract an average line bias

noao.imred.ccdred:

badpiximage	Create a bad pixel mask image from a bad pixel file
ccdgroups	Group CCD images into image lists
ccdheadit	CCD image header editor
ccdinstrument	Review and edit instrument translation files
ccdlist	List CCD processing information
ccdproc	Process CCD images
ccdtest	CCD test and demonstration package
combine	Combine CCD images
cosmicrays	Detect and replace cosmic rays
darkcombine	Combine and process dark count images
flatcombine	Combine and process flat field images
mkfringecor	Make fringe correction images from sky images
mkillumcor	Make flat field illumination correction images
mkillumflat	Make illumination corrected flat fields

mkskycor	Make sky illumination correction images
mkskyflat	Make sky corrected flat field images
setinstrument	Set instrument parameters
zerocombine	Combine and process zero level images

ADDITIONAL HELP TOPICS

ccdgeometry	Discussion of CCD coordinate/geometry keywords
ccdtypes	Description of the CCD image types
flatfields	Discussion of CCD flat field calibrations
guide	Introductory guide to using the CCDRED package
instruments	Instrument specific data files
package	CCD image reduction package
subsets	Description of CCD subsets

noao.imred.ccdred.ccdtest:

artobs	Create an artificial CCD observation
demo	Run a demonstration of the CCD reduction package
mkimage	Make or modify an image with simple values
subsection	Create an artificial subsection CCD observation

noao.imred.ctioslit:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
aprecenter	Recenter apertures
apresize	Resize apertures
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plot of spectra with SPLIT
calibrate	Apply extinction and flux calibrations to spectra
continuum	Fit and normalize the continuum of multispec spectra
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
identify	Identify arc lines and determine a dispersion function
refspectra	Assign reference spectra to object spectra
reidentify	Reidentify arc lines and determine new dispersion
functions	
sarith	Spectrum arithmetic
scombine	Combine spectra
scopy	Copy spectra including aperture selection and format
changes	
sensfunc	Create sensitivity function
setairmass	Compute effective airmass and middle UT for an exposure
setjd	Compute and set Julian dates in images
slist	List spectral header elements
specplot	- Stack and plot multiple spectra
splot	Plot and analysis spectra
standard	Identify standard stars to be used in sensitivity calc
doslit	Process CTIO slit spectra
demos	Demonstrations and tests

noao.imred.dtoi:

dematch	Match a list of density values to exposure values
hdffit	Fit a curve to density, log exposure values
hdshift	Align related HD curves
hdtoi	Apply DTOI transformation to density image
selftest	Self test program to check DTOI transformation
spotlist	Generate a list of calibration spot values

noao.imred.echelle:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters and apidtable
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
apfit	Fit 2D spectra and output the fit, difference, or ratio
apflatten	Remove overall spectral and profile shapes from flat
fields	
apmask	Create an IRAF pixel list mask of the apertures
apnormalize	Normalize 2D apertures by 1D functions
aprecenter	Recenter apertures
apresize	Resize apertures
apscatter	Fit and subtract scattered light
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plots of spectra
calibrate	Apply extinction and flux calibrations to spectra
continuum	Fit the continuum in spectra
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
ecidentify	Identify features in spectrum for dispersion solution
ecreidentify	Automatically reidentify features in spectra
refspectra	Assign wavelength reference spectra to other spectra
sarith	Spectrum arithmetic
scombine	Combine spectra
scopy	Select and copy apertures in different spectral formats
sensfunc	Compute sensitivity function
setairmass	Compute effective airmass and middle UT for an exposure
setjd	Compute and set Julian dates in images
slist	List spectrum header parameters
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
standard	- Identify standard stars to be used in sensitivity calc
doeslit	Process Echelle slit spectra
dofoe	Process Fiber Optic Echelle (FOE) spectra
demos	Demonstrations and tests

noao.imred.generic:

background	Fit and subtract a line or column background
cosmicrays	Detect and replace cosmic rays
darksub	Scale and subtract a dark count image
flat1d	Make flat field by fitting a 1D func. to the lines or
columns	
flatten	Flatten images using a flat field
normalize	Normalize images
normflat	Create a flat field by normalizing and replacing low
values	

noao.imred.hydra:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
aprecenter	Recenter apertures
apresize	Resize apertures
apscatter	Fit and remove scattered light
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plots of spectra

continuum	Fit the continuum in spectra
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
identify	Identify features in spectrum for dispersion
solution	
msresp1d	Create 1D response spectra from flat field and sky
spectra	
refspectra	Assign wavelength reference spectra to other spectra
reidentify	Automatically identify features in spectra
sapertures	Set or change aperture header information
sarith	Spectrum arithmetic
scombine	Combine spectra having different wavelength ranges
scopy	Select and copy apertures in different spectral
formats	
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
slist	List spectrum header parameters
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
dohydra	Process HYDRA spectra
demos	Demonstrations and tests

noao.imred.iids:

addsets	Add subsets of strings of spectra
batchred	Batch processing of IIDS/IRS spectra
bplot	Batch plots of spectra
bswitch	Beamswitch strings of spectra to make objsky pairs
calibrate	Apply sensitivity correction to spectra
coefs	Extract mtn reduced coefficients from henear scans
coincor	Correct spectra for detector count rates
continuum	Fit the continuum in spectra
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
extinct	Use BSWITCH for extinction correction
flatdiv	Divide spectra by flat field
flatfit	Sum and normalize flat field spectra
identify	Identify features in spectrum for dispersion
solution	
lcalib	List calibration file data
mkspec	Generate an artificial spectrum
names	Generate a list of image names from a string
powercor	Apply power law correction to mountain reduced
spectra	
process	A task generated by BATCHRED
refspectra	Assign reference spectra to object spectra
reidentify	Automatically identify features in spectra
scombine	Combine spectra having different wavelength ranges
sensfunc	Create sensitivity function
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
sinterp	Interpolate a table of x,y pairs to create a
spectrum	
slist1d	List spectral header elements
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
standard	Identify standard stars to be used in sensitivity
calc	
subsets	Substract pairs in strings of spectra
sums	Generate sums of object and sky spectra by aperture

noao.imred.irred:

center	Compute accurate centers for a list of objects
centerpars	Edit the centering parameters
datapars -	Edit the data dependent parameters
flatten	Flatten images using a flat field
iralign	Align the image produced by irmosaic
irmatch1d	Align and intensity match the image produced by
irmosaic (1D)	
irmatch2d	Align and intensity match the image produced by
irmosaic (2D)	
irmosaic	Mosaic an ordered list of images onto a grid
mosproc	Prepare images for quick look mosaicing
txdump	Select fields from the center task output text file

noao.imred.irs:

addsets	Add subsets of strings of spectra
batchred	Batch processing of IIDS/IRS spectra
bplot	Batch plots of spectra
bswitch	Beamswitch strings of spectra to make objsky pairs
calibrate	Apply sensitivity correction to spectra
coefs	Extract mtn reduced coefficients from henear scans
continuum	Fit the continuum in spectra
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
extinct	Use BSWITCH for extinction correction
flatdiv	Divide spectra by flat field
flatfit	Sum and normalize flat field spectra
identify	Identify features in spectrum for dispersion
solution	
lcalib	List calibration file data
mkspec	Generate an artificial spectrum
names	Generate a list of image names from a string
process	A task generated by BATCHRED
refspectra	Assign reference spectra to object spectra
reidentify	Automatically identify features in spectra
scombine	Combine spectra having different wavelength ranges
sensfunc	Create sensitivity function
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
sinterp	Interpolate a table of x,y pairs to create a
spectrum	
slist1d	List spectral header elements
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
standard	Identify standard stars to be used in sensitivity
calc	
subsets	Subtract pairs in strings of spectra
sums	Generate sums of object and sky spectra by aperture

noao.imred.kpnocoude:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
aprecenter	Recenter apertures
apresize	Resize apertures
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plot of spectra with SPLIT

calibrate	Apply extinction and flux calibrations to spectra
continuum	Fit and normalize the continuum of multispec
spectra	
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
identify	Identify arc lines and determine a dispersion
function	
msresp1d	Create fiber response spectra from flat field and
sky spectra	
refspectra	Assign reference spectra to observations
reidentify	Reidentify arc lines and determine new dispersion
functions	
sapertures	Set or change aperture header information
sarith	Spectrum arithmetic
scombine	Combine spectra
scopy	Copy spectra including aperture selection and
format changes	
sensfunc	Create sensitivity function
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
slist	List spectrum headers
specplot	Stack and plot multiple spectra
splot	Plot and analyze spectra
standard	Identify standard stars to be used in sensitivity
calc	
do3fiber	Process KPNO coude three fiber spectra
doslit	Process KPNO coude slit spectra
demos	Demonstrations and tests

noao.imred.kpnoslit:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
aprecenter	Recenter apertures
apresize	Resize apertures
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plot of spectra with SPLIT
calibrate	Apply extinction and flux calibrations to spectra
continuum	Fit and normalize the continuum of multispec
spectra	
deredden	Apply interstellar extinction corrections
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
identify	Identify arc lines and determine a dispersion
function	
refspectra	Assign reference spectra to observations
reidentify	Reidentify arc lines and determine new dispersion
functions	
sarith	Spectrum arithmetic
scombine	Combine spectra
scopy	Copy spectra including aperture selection and
format changes	
sensfunc	create sensitivity function
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
slist	List spectrum headers
specplot	Stack and plot multiple spectra
splot	Plot and analyze spectra

standard	Identify standard stars to be used in sensitivity
calc	
doslit	Process slit spectra
demos	Demonstrations and tests
noao.imred.specred:	
apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters and apidtable
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures
apfit	Fit 2D spectra and output the fit, difference, or
ratio	
apflatten	Remove overall spectral and profile shapes from
flat fields	
apmask	Create and IRAF pixel list mask of the apertures
apnormalize	Normalize 2D apertures by 1D functions
aprecenter	Recenter apertures
apresize	Resize apertures
apscatter	Fit and subtract scattered light
apsum	Extract 1D spectra
aptrace	Trace positions of spectra
bplot	Batch plot of spectra with SPLIT
calibrate	Extinction and flux calibrate spectra
continuum	Fit the continuum in spectra
deredden	Apply interstellar extinction correction
dispcor	Dispersion correct spectra
dopcor	Doppler correct spectra
fitprofs	Fit gaussian profiles
identify	Identify features in spectrum for dispersion
solution	
msresp1d	Create 1D response spectra from flat field and sky
spectra	
refspectra	Assign wavelength reference spectra to other
spectra	
reidentify	Automatically reidentify features in spectra
sapertures	Set or change aperture header information
sarith	Spectrum arithmetic
scombine	Combine spectra
scopy	Select and copy apertures in different spectral
formats	
sensfunc	Compute instrumental sensitivity from standard
stars	
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
sfit	Fit spectra and output fit, ratio, or difference
skysub	Sky subtract extracted multispec spectra
slist	List spectrum header parameters
specplot	Scale, stack, and plot multiple spectra
splot	Preliminary spectral plot/analysis
standard	Tabulate standard star counts and fluxes
dofibers	- Process fiber spectra
doslit	Process slit spectra
noao.imred.vtel:	
destreak	Destreak He 10830 grams.
destreak5	First pass processing CL script for 10830 grams.
dicoplot	Make dicomed plots of carrington maps.
fitslogr	Make a log of certain header parameters from a
FITS tape.	
getsqib	Extract the squibby brightness image from a full
disk scan.	

makehelium	Cl script for processing destreaked 10830
grams(second pass).	
makeimages	Cl script for processing magnetograms into
projected maps	
merge	Merge daily grams into a Carrington map.
mrotlogr	Log some header parameters from a FITS rotation
map tape.	
mscan	Read all sector scans on a tape and put them into
images.	
pimtext	Put text directly into images using a pixel font.
putsqib	Merge a squibby brightness image into a full disk
image.	
quickfit	Fit an ellipse to the solar limb.
readvt	Read a full disk tape and produce an IRAF image.
rmap	Map a full disk image into a 180 by 180 flat
image.	
syndico	Make dicomed print of daily grams 18 cm across.
tcopy	Tape to tape copy routine.
trim	Set all pixels outside the limb to 0.0 (use sqib
for limb).	
unwrap	Remove effects of data wraparound on continuum
scans.	
vtblink	Blink daily grams on the IIS to check for
registration.	
vtexamine	Examine a vacuum telescope tape, print headers and
profile.	
writetape	Cl script to write 5 full disk grams to tape.
writevt	Write an IRAF image to tape in vacuum telescope
format.	

noao.mtlocal:

ldumpf	List the permanent files on a Cyber DUMPF tape
r2df	Convert a CTIO 2d frutti image into an IRAF image
rcamera	Convert a CAMERA image into an IRAF image
rdumpf	Convert IPPS rasters from a DUMPF tape to IRAF
images	
ridsfile	Convert IDSFILES from a DUMPF tape to IRAF images
ridsmtn	Convert mountain format IDS/IRS data to IRAF
images	
ridsout	Convert a text file in IDSOUT format to IRAF
images	
rpds	Convert a PDS image into an IRAF image
rrcopy	Convert IPPS rasters from an RCOPY tape to IRAF
images	
widstape	Convert ONEDSPEC spectra to IDSOUT text format

noao.nproto:

binpairs	Bin pairs of (x,y) points in log separation
findgain	Estimate the gain and readnoise of a CCD
findthresh	Estimate a CCD's sky noise from the gain and
readnoise	
iralign	Align the mosaiced image produced by irmosaic
irmatch1d	Align and intensity match image produced by
irmosaic (1D)	
irmatch2d	Align and intensity match image produced by
irmosaic (2D)	
irmosaic	Mosaic an ordered list of images onto a grid
linpol	Calculate polarization frames and Stoke's
parameters	
slitpic	Generate IRAF image of aperture slit mask

noao.onedspec:

bplot	Batch plots of spectra
calibrate	Apply extinction and flux calibrations to spectra
continuum	Fit the continuum in spectra
deredden	Apply interstellar extinction correction
dispaxis	Dispersion axis parameters for 2D images
dispcor	Dispersion correct spectra
dopcor	Apply doppler corrections
fitprofs	Fit gaussian profiles
identify	Identify features in spectrum for dispersion
solution	
lcalib	List calibration file data
mkspec	Generate an artificial spectrum
names	Generate a list of image names from a string
ndprep	Make neutral density filter calibration image
refspectra	Assign wavelength reference spectra to other
spectra	
reidentify	Automatically identify features in spectra
sapertures	Set or change aperture header information
sarith	Spectrum arithmetic
scombine	Combine spectra having different wavelength ranges
scopy	Select and copy apertures in different spectral
formats	
sensfunc	Create sensitivity function
setairmass	Compute effective airmass and middle UT for an
exposure	
setjd	Compute and set Julian dates in images
sfit	Fit spectra and output fit, ratio, or difference
sinterp	Interpolate a table of x,y pairs to create a
spectrum	
slist	List spectrum header parameters
specplot	Stack and plot multiple spectra
splot	Preliminary spectral plot/analysis
standard	Identify standard stars to be used in sensitivity
calc	

ADDITIONAL HELP TOPICS

package sections on: Discussion and overview of package including spectral formats, dispersion coordinates, and units

noao.rv:

continpars	Edit continuum subtraction parameters
filtpars	Edit the filter function parameters
fxcor	Radial velocities via Fourier cross correlation
keywpars	Translate the image header keywords used in RV

package

rvcorrect	Compute radial velocity corrections
-----------	-------------------------------------

noao.twodspec:

apextract	Aperture Extraction Package
longslit	Longslit Package

noao.twodspec.apextract:

apall	Extract 1D spectra (all parameters in one task)
apdefault	Set the default aperture parameters and apidtable
apdemos	Various tutorial demonstrations
apedit	Edit apertures interactively
apfind	Automatically find spectra and define apertures

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

apfit	Fit 2D spectra and output the fit, difference, or ratio
apflatten	Remove overall spectral and profile shapes from flat fields
apmask	Create and IRAF pixel list mask of the apertures
apnormalize	Normalize 2D apertures by 1D functions
aprecenter	Recenter apertures
apresize	Resize apertures
apscatter	Fit and subtract scattered light
apsum	Extract 1D spectra
aptrace	Trace positions of spectra

ADDITIONAL HELP TOPICS

apbackground	Background subtraction algorithms
aprofiles	Profile determination algorithms
apvariance	Extractions, variance weighting, cleaning, and noise model
package	Package parameters and general description of package

noao.twodspec.longslit:

background	Fit and subtract a line or column background
extinction	Apply atmospheric extinction corrections to images
fitcoords	Fit user coordinates to image coordinates
fluxcalib	Apply flux calibration to images
identify	Identify features
illumination	Determine illumination calibration
reidentify	Reidentify features
response	Determine response calibration
setairmass	Compute effective airmass and middle UT for an exposure
setjd	Compute and set Julian dates in images
transform	Transform longslit images to user coordinates

Many of the tasks in the NOAO packages are visible in several different packages. Although these are the same tasks and provide the same functionality, the default parameters are often set differently to account for different instruments or uses.

EK C

BAZI TEMEL UNIX KAVRAMLARI ve KOMUTLARI

UNIX Nedir?

UNIX, bir çok kullanıcı ve zaman paylaşımı işletim sistemidir.

1970'li yılların başında AT&T laboratuvarlarında geliştirildikten sonra UNIX, üniversiteler, araştırma enstitüleri, devlet kurumları ve bilgisayar şirketleri tarafından, bugünün teknolojilerini geliştirmek üzere kullanılmaya başlandı. Bu teknolojiler bugün bir UNIX işletim sisteminin birer parçası veya bileşenidir.

Bilgisayar destekli tasarım (CAD), üretim kontrol sistemleri, laboratuvar simülasyonları ve Internet'in kendisi bile "yaşamlarına" UNIX ile ve UNIX olduğu için başladılar.

UNIX'te Dizinler ve İçerikleri

/	kök (root) dizin
/bin	önemli programların dizini ("binaries")

/dev	aygıt (tape, yazıcı vs.) dosyaları dizini
/etc	işletim sistemi ile ilgili çeşitli program dosyaları dizini
/etc/motd	günün açılış mesajının bulunduğu dizin
/etc/passwd	şifre dosyasının bulunduğu dizin
/lib	önemli kütüphanelerin bulunduğu dizin
/tmp	geçici ve sistemin açılışında silinen dosyalar dizini
/unix	işletim sistemi programının içinde bulunduğu dizin
/usr/adm	sistem yönetimi ile ilgili dosyaların bulunduğu dizin
/usr/bin	kullanıcı programlarının dizini (troff, vb.)
/usr/dict	sözlük (words) ve spell(1) için destek dizini
/usr/games	oyun programları dizini
/usr/include	C programlarının başlık dosyaları (math.h vb.) dizini
/usr/include/sys	C programlarının sistem başlık dosyaları (inode.h vb.) dizini
/usr/lib	C, FORTRAN vb. için kütüphaneler dizini
/usr/man	on-line el kitabı dizini
/usr/man/man1	on-line el kitabının 1. Bölümünün bulunduğu dizin
/usr/mdec	donanım diagnostikleri, bootstrap programları vb.
/usr/src	hizmet programları ve kütüphaneler için kaynak kodlar dizini
/usr/src/cmd	/bin ve /usr/bin dizinlerindeki komutların kaynak kod dizini
/usr/src/lib	altprogram kütüphanelerinin kaynak kod dizini
/usr/spool	iletişim programlarının çalışma dizini
/usr/spool/lpd	satır yazıcısının geçici dizini
/usr/spool/mail	mail gelen kutuları dizini
/usr/spool/uucp	uucp programlarının çalışma dizini
/usr/sys	işletim sistemini çalıştıran kaynak kod dizini
/usr/tmp	alternatif bir dizin (çok az kullanılır)
/usr/you	login dizini (her kullanıcının böyle bir dizini vardır)
/usr/you/bin	her kullanıcının kişisel programları

UNIX Komutlarının Kısa Tanımı

alias	aliasları sıralama veya yeni alias tanımlama
awk	belirli ifadeleri (pattern) tarama ve işleme
a.out	program çıktısı dosyası
cal	takvim çağırma
calendar	hatırlatma servisi
cat	birleştirme ve azdırma
cc	C derleyicisini çalıştırma
cd, chdir	dizin değiştirme
chmod	dosya modlarını/haklarını değiştirme
cp	dosyaların kopyasını alma
date	tarih ve zaman bildirir
emacs	emacs editörünü çalıştırma
ex	satır esaslı editör programını çalıştırır
f77	FORTRAN programlarını derleme
find	belirli bir dosyayı bulma ve üzerinde bir işlem gerçekleştirme
finger	kullanıcılar hakkında bilgi verme
grep	bir dosyanın içinde belirli bir ifadeyi (paterni) arama
head	bir dosyanın başlığına bakma
history	son işletilen komutların listeleme
jobs	durdurulmuş ve arka planda çalışan tüm işleri listeleme
join	iki sütünü yatay olarak birleştirme
kill	çalışan işleri (programları) durdurma
learn	UNIX ile ilgili bilgi verme
ln	dosya linkleri oluşturma
login	sisteme kullanıcı olarak girme

lpr	satır yazıcıyı kullanma
ls	bir dizinin içeriğini listeleme
mail	mail programını çalıştırma (gelen ve giden mail için)
man	anahtar kelimelere göre on-line bilgi alma
mesg	"mesaj al" veya "mesaj alma" konumuna getirme
mkdir	dizin yaratma
more	uzun dosyaları sayfa sayfa (ekran) listeleme
mv	dosyaları silme veya adını değiştirme
nroff	metin formatlama
passwd	login şifresini değiştirme
pc	Pascal programlarını derleme
pr	kısmen formatlanmış bir dosyayı yazdırma
ps	durum raporu verme
pwd	çalışılan dizini gösterme
rm	dosyaları silme
rmdir	dizinleri silme
sort	dosyaları sıralar ve birleştirir
spell	heceleme denetçisi
stty	terminal seçeneklerini ayarlama
tail	bir dosyanın son bölümünü gösterme
talk	başka bir kullanıcı ile konuşma
te	e çıktığı bölme
time	bir komutun zamanını verme
uniq	bir dosyadaki tekrarlanan satırları siler
vi	monitör esaslı metin editörünü çalıştırır
wc	kelimelerin adedini verir
who	sistemde çalışanları gösterir

EK D

vi EDITÖRÜ KOMUTLARI

vi Editörü ve Kullanımı Hakkında Kısa Bilgi

UNIX'in ilk editör programı '**ed**' isimli ve satır esaslı çalışan bir editördü. Daha sonra University of California Berkeley'de geliştirilen '**vi**' editörü '**ed**' 'in ve onun kadar eski olan '**ex**' editörünün yerini aldı.

'**vi**' editörünün 'Insert', 'Escape' (command) ve 'Last-Line' olmak üzere üç farklı modu vardır. 'Insert' modunda klavyeniz yardımı ile yazma işini gerçekleştirirsiniz. Bu mod sizin satırlar arasında aşağı yukarı gidip gelmenize müsaade etmez. 'Escape' modunda ise her harf ya da sayının bir işlevi vardır. Bu sayede metnin sonuna ve başına gidilebilir. Bu modda iken a,A,i,I,o ve O harflerini kullanarak 'Insert' moduna geçebilirsiniz. 'Escape' modunda iken '.' tuşuna basıldığında 'Last-Line' moduna geçilir. Burada metin içinde ekleme, çıkarma, değiştirme gibi temel işlevleri yerine getirebildiğiniz gibi, metnin bir kısmını başka adla diske yazabilir, farklı bir dosyayı metne katabilirsiniz.

'**vi**' kullanırken yaptığınız tüm değişiklikler buffer üzerinde olduğundan, yapılan değişikliklerin kalıcı olması isteniyorsa 'Last-Line' modunda 'w' (write) komutu kullanılmalıdır.

Cursor'ın Hareketini Sağlayan Komutlar

j	bir satır aşağıya
k	bir satır yukarıya
h	bir karakter sola

l bir karakter sağa
Ctrl-d ekranı yarım sayfa aşağıya
Ctrl-u ekranı yarım sayfa yukarıya
Ctrl-b ekranı bir sayfa geriye
Ctrl-f ekranı bir sayfa ileriye
nG cursoru n. satıra

Metin Ekleme Komutları

a cursordan sonra
i cursordan önce
o cursordan sonra satır ilave eder
O cursordan önce satır ilave eder

Metin Silme Komutları

x cursorun altındaki karakteri siler
dw cursordan bir sonraki kelimeye kadar siler
dd cursorun üzerindeki satırı siler
d) cümlelerin kalanını siler
d} paragrafın kalanını siler

Not: Bu komutlarla birlikte sayı da kullanılabilir

Metin Değiştirme Komutları

r cursorun altındaki karakteri girilen karakterle değiştirir
R cursordan başlayarak eski metnin üzerine yazar
cw kelimeyi yeni metinle değiştirir
c) cümleyi yeni metinle değiştirir
J üstteki satırı cursorun üzerinde bulunduğu satırla birleştirir
u son uygulanan işlemi iptal eder
U cursorun üzerinde bulunduğu satırdaki tüm değişiklikleri iptal eder

Not: R, cw, c) komutlarından sonra (Esc) tuşuna basılmalıdır.

Arama Komutları

/ifade ifade ile uyuşanları arar
ifade ifade'nin daha önce geçtiği yeri araştırır
n son arama komutunu tekrarlar

Son Komut ile İlgili İşlemler

u son komutu iptal eder
. son komutu tekrar eder
U satırdaki tüm değişiklikleri iptal eder

Metin Hareketini Sağlayan Komutlar

yy bir satırın kopyasını alıp geçici belleğe yerleştirir
p yy ile en son kopyalanan veya silinen bölümü cursordan sonrasına yerleştirir
P bir önceki işlemin tersini yapar
"cY bir satırın kopyasını alır, onu c isimli geçici bellek bölgesine yerleştirir
(c, burada a'dan z'ye herhangi bir harf olabilir)
"cP c geçici bellek bölgesinin içeriğini cursordan sonrasına yerleştirir

Metni Kaydetmek ve Editörden Çıkmak

<Esc>:w yazılan metni kalıcı belleğe yerleştirir

```
<Esc>:q          metinde deęişiklik yapmadan editörü terketmek
<Esc>:q!        Önceki komutun daha etkili hali
<Esc>:wq        kaydet ve çık
<Esc>:ZZ        kaydet ve çık
<Esc>:n, kw dosya2  n'den k'ya kadar olan satırları
                  dosya2 isimli dosyaya kaydeder
<Esk>:n, kw >> dosya2  n'den k'ya kadar olan satırları
                  dosya2 isimli dosyaya ilave eder
```

KAYNAKLAR

- Barnes J., “A Beginner’s Guide to Using IRAF – IRAF Version 2.10” [Draft],NOAO, August 1993
- Kernighan B., Pike R., The UNIX Programming Environment, 1984
- McLean I.S., Electronic Imaging in Astronomy-Detectors and Instrumentation, 1997
- Tody D., “The IRAF Data Reduction and Analysis System”, NOAO
- Tody D., “IRAF in the Nineties”, NOAO
- Waite M., Martin D., ve Prata S., The Waite’s Goup’s UNIX Primer Plus, Sec. Ed., 1995
- Astronomical CCD Observing and Reduction Techniques, ASP Conference Series, Vol. 23, 1992, Steve B. Howell, ed.
- STSDAS User’s Guide, Space Telescope Science Institute, April 1994

ACKNOWLEDGMENTS

We hope that this guide will be useful for all Turkish scientists who plan to use IRAF for data reduction and analysis. For the most part, this guide is a translation of the original one written by Jeannette Barnes (A Beginner's Guide to Using IRAF), but we have added some additional information about Unix at the end. We wish to thank Jeannette Barnes who arranged for our visit to NOAO, and our special thanks to the members of the IRAF group and the Central Computer Services department who worked with us during our visit, Lindsey Davis, Dave Bell, Mike Fitzpatrick, Frank Valdes, Rob Seaman, Nelson Zarate, and Mike Peralta. We will miss you! We would also like to thank Daryl Willmarth who provided us with several Kitt Peak tours and with some sample data for IRAF data reduction. We wish to thank as well all those who just smiled at us during our stay and made NOAO a warm and comfortable place to study.

TESEKKÜR

[NOAO \(National Optical Astronomy Observatories\)](#)'ya gelmeden bu çalışmanın gerçekleştirilmesi mümkün olmazdı. Buraya gelme ve çalışma fikrini bize veren ve desteğini esirgemeyen [bölüm](#) başkanımız Prof.Dr. Dursun Koçer'e çok teşekkür ederiz.

NOAO'ya gelmemizi sağlayan ve bize IRAF ekibi ile tam donanımlı bir ortamda çalışma ortamı yaratan sayın Jeannette Barnes'e teşekkürü bir borç biliriz. Ayrıca IRAF ekibinin bize yardımcı olan tüm elemanlarına – Lindsey Davis, Dave Bell, Mike Fitzpatrick, Frank Valdes, Rob Seaman, Nelson Zerate ve Mike Peralta – yardımseverlikleri, sonsuz sabırları ve yakınlıkları için ayrı ayrı teşekkür ederiz.

Bize Kit Peak Gözlemevini tanıtan ve orada iki harika gün gecirmemizi sağlayan Daryl Willmarth'a da teşekkürlerimizi sunarız.

Bu çalışmanın kısa bir sürede gerçekleştirilmesinde Compaq Computer Ticaret A.S.

Yrd. Doç. Dr. Sevinç Gülseçen, Yrd. Doç. Dr. Hulusi Gülseçen, 3.Mart.2005

Kurumsal Cözümler ve Hizmetler Grup Müdürü sayın Hakan Arıtürk'ün şüphesiz çok büyük bir katkısı olmuştur, ona da minnettarız.

Ayrıca İ.Ü. Rektörlüğü, İ.Ü. Fen Fakültesi Dekanlığı ve İ.Ü. Gözlemevi Araştırma ve Uygulama Merkezi Müdürlüğü'ne de teşekkür ederiz.

Bu dökümanı HTML formatına dönüştürülmesini gerçekleştiren öğrencilerimiz, F. Özgür Mülazım ve Z. Burçak Kanatsız'a teşekkürlerimizi sunarız.

Dileğimiz bu çalışmanın, IRAF'ı kullanacak olan herkese yardımcı olmasıdır.