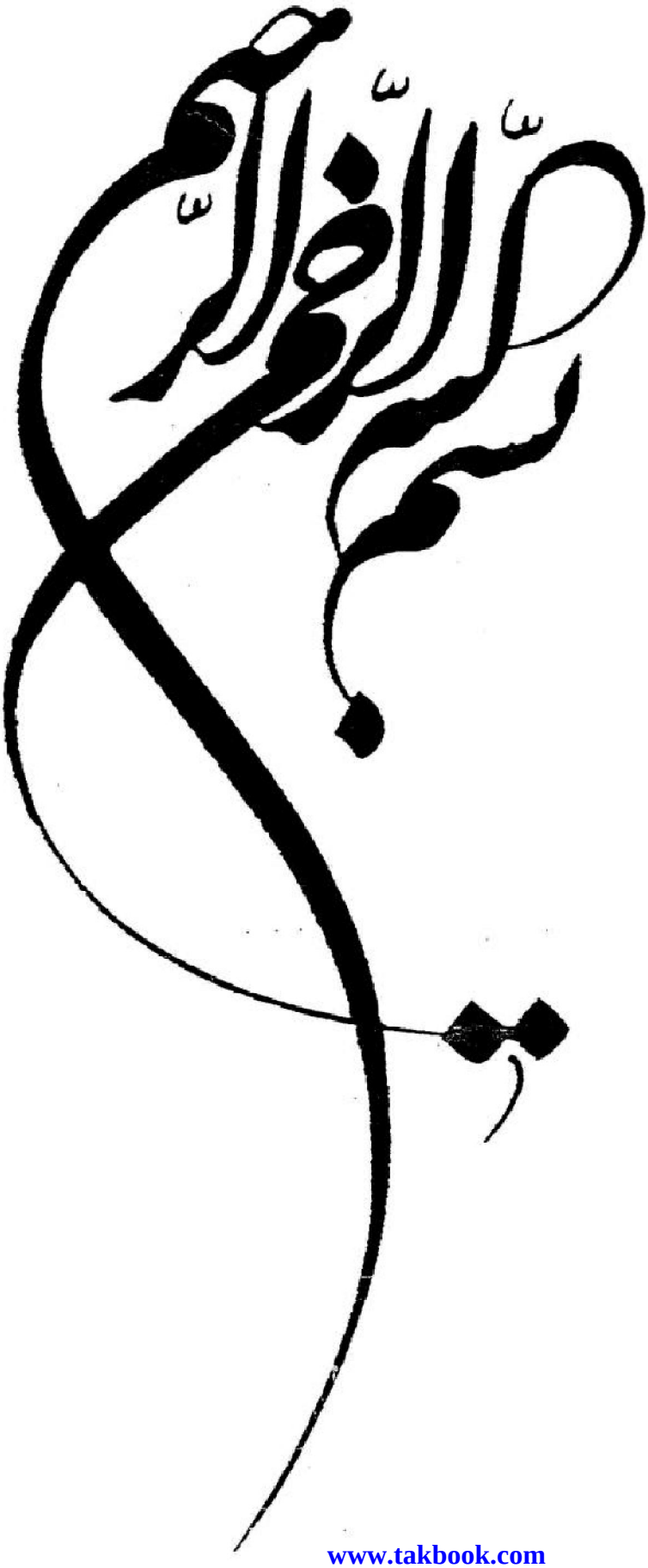
2014

زمان بندی فرآیندها توسط Cron

نویسنده : حسام الدین توحید

SKYWAN13@YAHOO.COM



مقدمه مولف :

آنچه پیش رو دارید به صورت رایگان و تحت لیسانس GNU GPLv3 به علاقه مندان لینوکس هدیه می گردد . در تهیه این مقاله از سر فصل های درسی گفته شده در دوره هایLPIc2 و RHCE استفاده شده ولازم می دانم از مهندس مهدوی فر به خاطر راهنمایی های مفیدشان و مرکز آموزشهای پیشرفته دانشگاه شریف (لایتک) تشکر کافی را داشته باشم. این مطالب با نگاهی کاربردی و بدون پرداختن به بحث های تئوریک گردآوری و عرضه شده است. امیدوارم مطالب ارائه شده بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. این آموزش بر اساس توزیع CentOSمی باشد، لذا خواهشمندم هرگونه نقص در محتوا را به ایمیل نگارنده ارسال فرمائید. انتشار این فایل با ذکر منبع بلامانع است و هر گونه استفاده نامناسب از محتوای ارائه شده بر عهده کاربر بوده و تمام حقوق معنوی این اثر به نویسنده آن تعلق دارد.

زکات علم نشر آن است.

موفق باشید

حسام الدین توحید

تیر 1393

فهرست مطالب :

زمان بندی اجرای برنامه ها توسط Cron

مقدمه :

Cron بر گرفته از chromos یک کلمه یونانی به معنای زمان است که به خدای زمان یونان باستان گفته می شد و در یونیکس و سولاریس برای خودکارسازی انجام دستور ها و پروسس ها از آن استفاده می شود.
برنامه ریزی و زمانبندی برای انجام فعالیت های مختلف در لینوکس امر بسیار مهمی هست که شاید خیلی از ما روزانه با آن سر و کار داشته باشیم. برخی اوقات ما دستور یا دستورات مورد نظر خود را به طور مستقیم از شل میخواهیم و انجام میشود اما برخی اوقات نیاز هست تا در زمان (یا زمان هایی) مقرر سیستم عامل به طور خودکار برای ما کاری را انجام دهد. برای مثال نیاز داریم تا سیستم برای ما هرروز در ساعتی که مشخص میکنیم یکاپ بگیرد. از مهمترین راه های زمانبندی برنامه ها در سیستم عامل های شبه یونیکس استفاده از نرم افزار cron و دستور at می باشد. زمانی که به صورت یه دوره متناوب بخواهیم فعالتی انجام شود میتوانید از Crond و Cron Table ها کمک بگیریم.
مدیریت زمانبندی اجرای دستورات توسط این برنامه در قالب یک فایل به نام crontab که بر گرفته از crontable است و معمولا در مسیر /etc/crontab/ قرار دارد انجام می شود. وقتی ما خطی را به فایل crontab اضافه می کنیم ، برای اعمال شدن آن حتما باید سرویس cron ریست شود تا cron مجبور به خواندن فایل پیکربندی شده و اسکریپت جدید را در صف اجرا قرار دهد.
همچنین هرکاربر دارای یک فایل شخصی cron است که در مسیر var/spool/cron قرار دارد. پروسه cron هر یک دقیقه یکبار به فایل crontab رجوع کرده و از آن stat می گیرد ، اگر تغییری در این فایل اعمال شده باشد یکبار از اول این فایل را می خواند.

نصب و راه اندازی سرویس cron

ابتدا باید از نصب بودن پکیج cron اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم. در سری 6 به بعد CentOS نام این سرویس به cronie تغییر یافته است:

```
#rpm  -qa | grep  cronie
```

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب cron از yum استفاده می کنیم :

```
#yum  –y  install  cronie
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج cron بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

```
#rpm  -qa | grep  cronie
```

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm  -ql  cronie
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm  -qi  cronie
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
# chkconfig  --level  35  crond  on
```

و در انتها سرویس را reset می کنیم :

```
#service  crond  restart
```

مسیرها و فایل های اضافه شده به سیستم

با نصب این سرویس چندین مسیر مهم در سیستم اضافه می شود که در زیر مختصرا توضیح داده می شود :

/etc/cron.d
/etc/pam.d/crond
/etc/rc.d/init.d/crond
/etc/sysconfig/crond
/etc/crontab
/usr/bin/crontab

/etc/cron.d : در این دایرکتوری محتوایی وجود ندارد. فایل های cron ای که می سازیم در این دایرکتوری قرار می گیرد.

/etc/pam.d/crond : مکانیزم احراض هویت cron توسط pam اجرا می شود. در این دایرکتوری فایل کانفیگ آن قرار دارد.

/etc/rc.d/init.d/crond : سرویس cron زیر مجموعه init اداره میشود. در این مسیر اسکریپت startup سرویس cron قرار دارد.

/etc/sysconfig/crond : اگر بخواهیم سرویس cron با یکسری فیچر خاص استارت شود باید فیچرهای مورد نظر را در این فایل قرار دهیم. مثل debugging mod و یا اجرای سرویس روی یک کارت شبکه خاص .

/etc/crontab : در این مسیر فایل اصلی پیکربندی cron قرار دارد.

/usr/bin/crontab : و در این جا هم فایل دستور cron قرار گرفته است.

چندین فایل وجود دارد که برای تمامی سیستم بوده و مالک این فایل ها کاربر root است. همه کاربران دارای یک فایل crontab مخصوص خود هستند ولی فایلهای زیر متعلق به کل سیستم می باشند:

- etc/cron.d/ : دایرکتوری شامل چندین فایل
- etc/cron.daily/ : زمانبندی برای انجام روزانه

- زمانبندی بصورت ساعتی : /etc/cron.hourly
- زمانبندی ماهانه : /etc/cron.monthly
- زمانبندی هفتگی : /etc/cron.weekly

به طور مثال فایل هایی که در شاخه /etc/cron.daily/ قرار دارند- بطور روزانه اجرا خواهند شد. در زیر نمونه ای از محتویات این دایرکتوری را مشاهده میکنید:

```
# ls -l /etc/cron.daily/
total 56
-rwxr-xr-x 1 root root   ۳۱۱ Jun 20  ۲۰۱۰ ۰anacron
-rwxr-xr-x 1 root root 15399 Apr 20  ۲۰۱۲ apt
-rwxr-xr-x 1 root root   ۳۱۴ Mar 30  ۲۰۱۲ aptitude
-rwxr-xr-x 1 root root   ۵۰۲ Mar 31  ۲۰۱۲ bsdmainutils
-rwxr-xr-x 1 root root   ۲۵۶ Apr 13  ۲۰۱۲ dpkg
-rwxr-xr-x 1 root root   ۳۷۲ Oct  ۵  ۲۰۱۱ logrotate
-rwxr-xr-x 1 root root  ۱۳۶۵ Mar 31  ۲۰۱۲ man-db
-rwxr-xr-x 1 root root   ۶۰۶ Aug 17  ۲۰۱۱ mlocate
-rwxr-xr-x 1 root root   ۲۴۹ Apr  ۹  ۲۰۱۲ passwd
-rwxr-xr-x 1 root root   ۳۸۳ Apr 25  ۲۰۱۲ samba
-rwxr-xr-x 1 root root  ۲۹۴۷ Apr  ۲  ۲۰۱۲ standard
```

توضیح فایل پیکربندی سرویس Cron

سرویس cron در دو scope کار می کند : 1) Global 2) User Base و به طور کلی دو نوع فایل برای cron مورد استفاده قرار میگیرد User cron table & system cron table .

1) Global (system cron table): این نوع از کانفیگ مربوط به مدیریت پروسه های سرور است و ربطی به یوزرها نداشته و فقط در اختیار یوزر root می باشد. فایل پیکربندی گلوبال cron در مسیر /etc/crontab و فایل های سیستم نیز برای زمانبندی فعالیت ها در/etc/cron.d/ قرار دارد. نوشتن این نوع از تنظیمات فقط در اختیار یوزر root می باشد و یوزرهای عادی فقط اجازه دیدن فایل ها را دارند.

2) User Base (user cron table): ولی این نوع از کانفیگ را یوزرهای عادی سیستم برای انجام کارهای شخصی خودشان انجام می دهند. فایل های یوزر در آدرس /var/spool/cron میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود .

توضیحات نوشتن cron در بخش بعدی آمده است. فایل crontab ، فایل اصلی پیکربندی این سرویس می باشد تنظیمات global از طریق این فایل به سیستم اعمال می گردد. در ادامه به تشریح خطوط مهم این فایل می پردازیم :

#vi /etc/crotab

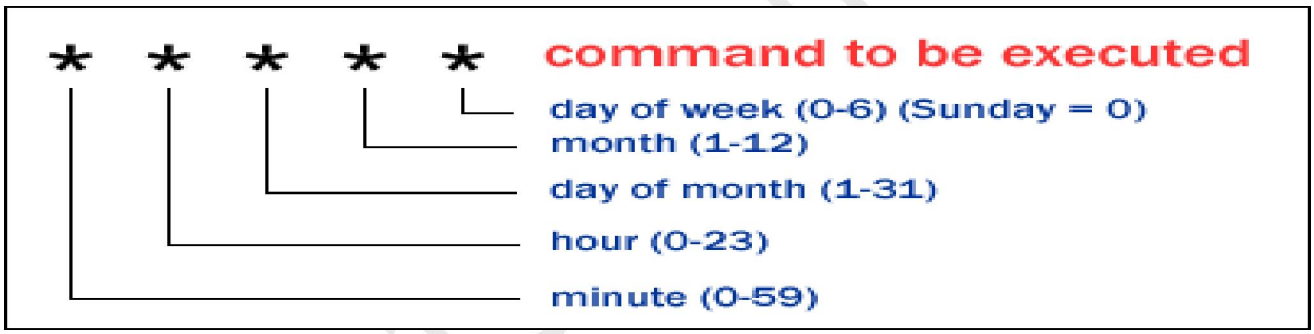
```
SHELL = /bin/bash
PATH = sbin/bin: /usr/sbin: /usr/bin
MAILTO = root
HOME = /
 *  *  *  *  *    root    run-parts    /etc/cron.daly
```

SHELL = /bin/bash: این مشخص می کند اسکریپتهای که می خواهیم با cron اجرا کنیم با چه شلی اجرا شوند.

PATH = sbin/bin: /usr/sbin: /usr/bin: یک اسکریپت مجموعه ای از دستورات است که با هم ترکیب شده اند. این خط مشخص می کند دستورات داخل اسکریپت از چه مسیرهایی اجرا شوند. اگر مسیر دستورات داخل اسکریپت در اینجا درج نشود آن بخش از اسکریپت و یا همه آن کار نمی کند.

MAILTO = root: به طور پیش فرض سرویس cron خروجی کار خود را بر روی صفحه نمایش نشان نمی دهد بلکه در حالت پیش فرض cron خروجی را برای email کاربر مالک job میفرستد. بهتر است به جای ایمیل شدن خروجی cron ، آن را به یک فایل redirect کنید تا دسترسی به آن راحت تر باشد. این دستور در صورتی اجرا خواهد شد که از /dev/null/ برای redirect استفاده نکرده باشید. اگر به آخر فایل مربوطه چیزی اضافه نشد یعنی اینکه خط نوشته شده ایراد دارد.

HOME = /: مسیر خانگی cron را نشان می دهد.



* * * * * root run-parts /etc/cron.daly/

این خط مهمترین قسمت این فایل است که دارای 8 فیلد می باشد:

Field 1 (Minuets) 0 – 59:

فیلد اول نشان دهنده دقیقه بوده و از 0 تا 59 متغیر است. محدودیت cron در دقیقه است. یعنی حداقل زمان بین دو کار یک دقیقه می باشد و نمی تواند به ثانیه کار کند.

Field 2 (Hour) 0 – 23:

فیلد دوم نشان دهنده ساعت بوده و از صفر تا 23 متغیر است.

Field 3 (Day of Month) 1 – 31:

این فیلد نشان دهنده روز از ماه می باشد.

Field 4 (Month) 1 – 12:

فیلد چهارم به ماه اشاره دارد.

Field 5 (Day of Week) 0 – 7:

فیلد پنجم مربوط به هفته می باشد. در cron یکشنبه برابر با صفر است و صفر با 7 فرقی ندارد لذا به جای 7 از 6 استفاده می شود.

نکته : ستاره (*) در cron معنی هر می دهد مثل هر ساعت یا هر دقیقه.

Field 6 (root):

این فیلد مشخص می کند اسکریپتی که مسیر آن را در اینجا مشخص کرده ایم توسط چه یوزری اجرا می شود.

Field 7 (run-parts):

اگر ما تعداد زیادی اسکریپت را درون یک دایرکتوری قرار دهیم و بخواهیم بوسیله cron اجرا شود تنها راه آن استفاده از دستور run-parts می باشد. برای اجرای یک مجموعه اسکریپت در زمان مشخص از run-parts استفاده میکنیم. این دستور اسکریپتهای یک دایرکتوری را به ترتیب حروف الفباء ، به صورت تک تک پشت سر هم اجرا می کند. این دستور با پکیج crontabs بر روی سیستم نصب می شود.

دستور `witch run-parts` مشخص میکند run-parts مربوط به چه پکجی است. #rpm -qf

Field 8 (/etc/cron.daily):

فیلد آخر هم اسکریپت اجرایی و مسیر آن را مشخص می کند.

نکته :

1) در خطوط cron نباید از دبل کوتیشن استفاده کنید.
2) با دستور زیر می توانیم از زمان آخرین تغییرات یک فایل مطلع شویم.

```
# stat   /etc/crontab
```

3) و با این دستور می توانیم log تغییرات crontab را مشاهده کنیم.

```
# tail   -f   /var/log/cron
```

برای ویرایش crontab یک کاربر، از طریق کاربر root بدین شکل عمل میکنیم:

```
#crontab -e -u username
```

استفاده از سرویس Cron

همانطور که گفته شد اسکوپ Global مخصوص یوزر root است و فقط root حق edit آن را دارد. یوزرهای عادی فقط اجازه دیدن فایل Global را دارند حال اگر یوزرهای عادی سیستم بخواهتد یک cron تعریف کنند باید از اسکوپ UserBase بهره ببرند. مکان ذخیره سازی فایل های cron یوزرها در آدرس /var/spool/cron/ میباشد و همانطور که از نام آن پیداست این فایل توسط کاربران ایجاد میشود. این دایرکتوری به طور پیش فرض خالی است و cron تعریف شده توسط یوزرها در اینجا ذخیره می شود. یوزر عادی برای تعریف cron باید از دستور crontab -e استفاده کند که editor پیش فرض آن ، vi است.

نکته مهمی که وجود دارد این است که در cron جدید دیگر نیازی نیست که تمام محتویات یک فایل cron را واردآن کنیم بلکه فقط خط مربوط به اجرای اسکریپت و زمانبندی را وارد می کنیم.

```
# crontab  -e
2  17  *  *  *  ls /
```

هر فایل cron ای که در /var/spool/cron/ ذخیره شود با نام یوزر سازنده آن ذخیره می شود و فقط همان یوزر و یوزر root حق خواندن و edit آن را دارند. یوزر root با دستور زیر می تواند cron متعلق به یک یوزر را edit کند.

```
# crontab  -u  skywan13  -e
```

اگر یوزری بخواهد چند cron داشته باشد باید تمام آنها را در یک فایل نوشته و ذخیره کند. اگر هم فیلدی را به اشتباه در فایل cron وارد کند در هنگام ذخیره به کاربر هشدار داده می شود. چک کردن فایل cron توسط crontab انجام می شود.

نکته مهم : یک سری علایم در نوشتن ورودی cron استفاده میشود که سعی شده با مثال توضیح داده شود:

* (ستاره) برای نادیده گرفتن یک فیلد در نظر گرفته شده، یعنی اگر مثلا در فیلد ساعت بود بدون توجه به این فیلد سر هر ساعت دستور اجرا میشود و یا اگر در فیلد دقیقه بود سر هر دقیقه و....

, (کاما) در هر فیلد که احتیاج داریم چندین بار دستور در ساعات مختلف اجرا بشود کاربرد دارد مثلا در فیلد دقیقه 15,30 به معنای اجرا در دقیقه های 15 و 30 یا در فیلد ساعت 2,9 به معنای اجرا در ساعتهای 9 و 2 میباشد و....

– (خط تیره) برای تعیین یک بازه زمانی است مثلا در فیلد روزهای ماه اگر بخواهیم بین روزهای 8 تا 15 دستور اجرا بشود بدین صورت مینویسیم 15-8 و....

همانطور که پیش تر در بالای همین مطلب ذکر شد، شما می توانید با دستور "crontab -e"یک فایل crontab بسازید. به هر حال ممکن است شما از قبل یک فایل crontab داشته باشید. برای مشخص کردن فایل خود، دستور زیر را وارد می کنیم :

```
crontab -u <username> <crontab file>
crontab  -u  skywan13  sky.log
```

سپس دستور زیر را وارد کنید تا فایل crontabکاربر skywan13 با نام crontab در پوشه آن خانگی آن ذخیره شود.

```
crontab –u  skywan13  ~/crontab
```

و برای حذف فایل crontab دستور زیر را در cli وارد می کنیم :

```
# crontab  -r
```

برای لیست کردن jobهایی که با crontab -e اضافه کردیم میتوانیم با استفاده از دستور زیر cron jobها را لیست شده ببینیم:

```
# crontab  -l
```

تا اینجا دیدید که برخلاف ظاهر پیچیده ، crontab به آسانی تنظیم میشود و ابزاری کاربردی و مهم در فرآیند مدیریت سیستم میباشد.

نکته مهم : برای استفاده از cron باید از crontab جهت load کردن jobها استفاده کرد که برای این منظور 2 راه پیش رو داریم :

۱-استفاده از crontab -e ,یعنی مستقیما دستور مربوط رو در crontab مینویسیم.

۲-از طریق فایل یعنی ابتدا یک فایل متنی میسازیم و طبق قوانین توضیح داده شده در بالا ورودی مناسب با شرایط خودتان رو در فایل مورد نظر قرار داده و سپس فایل را Load میکنید.

قبل از Load با استفاده از crontab -l لیست jobهای جاری را تهیه کرده و هر کدام را که لازم داریم در فایل جدید مینویسید چون با load این فایل تمامی jobهای گذشته پاک خواهند شد. جهت درک بهتر موضوع به مثال زیر توجه کنید:

```
#touch /skywan13/mycrontab
#echo "30  4  *  *  *    ls –s   / skywan13/web >> / skywan13/webdirlist.log
2>&1" > / skywan13/mycrontab
#crontab  -u  skywan13/  skywan13/mycrontab
#crontab -l
```

با اجرای crontab -l از load شدن فایل اطمینان حاصل میکنیم. توجه داشته باشید که برای افزودن job جدید یا ویرایشjob های موجود کافی است با ویرایش فایل mycrontab و افزودن دستور مورد نظر در خط جدید و پاک کردن crontab با استفاده از crontab -r فایل را دوباره Load کنیمcron jobهای تحت کاربری که فایل را با آن load کردیم اجرا خواهد شد.

محدود کردن دسترسی یوزرها برای استفاده از Cron

اگر بخواهیم برای استفاده از cron دسترسی یوزر و یا گروهی از یوزرها را محدود کنیم و یا فقط به بعضی از یوزرها دسترسی بدهیم باید در فایل های cron.deny و cron.allow تغییراتی اعمال کنیم. این دو فایل در زیر شاخه /etc قرار دارند. اگر هم وجود نداشتند مهم نیست چون می توانیم آنها را بسازیم.
اگر فایل cron.allow موجود باشد **حتما** باید اسامی یوزرهای مجاز در آن وارد شود، در غیر این صورت به هیچ یوزری استفاده از cron داده نمی شود.

اگر هم فایل cron.allow وجود نداشته باشد چک می شود آیا cron.deny وجود دارد یا خیر. اگر موجود باشد و نام یوزری در آن آمده باشد از دسترسی آن به cron جلوگیری می شود. اگر هیچ کدام از این دو فایل وجود نداشته باشد اجازه کار با cron به همه یوزرها داده می شود.

نکته مهم : چون ارجحیت اجرا با cron.allow است ابتدا این فایل مورد بررسی قرار می گیرد در صورت وجود نداشتن و یا خالی بودن آن ، فایل cron.deny مورد بررسی قرار می گیرد.

مثالهای برای Cron

برای درک بهتر ، مثال هایی به همراه توضیحات آورده شده است. ابتدا مثالهایی ساده بیان می شود و سپس مثال هایی پیشرفته مورد بررسی قرار می گیرد.
نوشتن فایل crontab ممکن است برای اولین بار کمی گیج کننده به نظر برسد. بنابراین در زیر تعدادی مثال ساده به همراه توضیح آورده شده است تا ابتدا با شیوه نوشتن cron آشنا شوید:

هر دقیقه اجرا می شوند<command> * * * * *

هر ساعت رأس دقیقه ۳۰ ام اجرا می شوند<command> * * * * 30

هر روز ساعت ۶:۴۵ صبح اجرا می شوند<command> * * * 6 45

هر روز صبح ساعت ۶:۴۵ بعد از ظهر اجرا می شوند<command> * * * 18 45

هر یکشنبه ساعت ۱ صبح(بامداد؟) اجرا می شوند<command> 0 * * 1 00

هر یکشنبه ساعت ۱ بامداد اجرا می شوند<command> 7 * * 1 00

هر یکشنبه ساعت ۱ بامداد اجرا می شوند<command> Sun * * 1 00

اولین روز هر ماه ساعت ۸:۳۰<command> * * 1 8 30

ماه پنجم هرروز در ساعت 5 هر دقیقه یکبار <command> 5 * 5 *

روز اول ماه اول سال <command> * 1 1 0 0

هر پنج دقیقه راس هر ساعت <command>* * * * */5

از دقیقه 5 هر 15 دقیقه به 15 دقیقه <command>* * * * 5/15

از روز سوم ، 3 روز به 3 روز <command>* * 3/3 * *

هر روز، هر ساعت دقیقه 20 و 50 <command>* * * 20,50

دقیقه 5، هر 3 ساعت یکبار روزهای اداری 1-5 * * */3 5

هر روز بین 5 الی 10 صبح <command>* * * 5,6,7,8,9,10 *

برای اجرای برنامه ها در startup از طریق crontab ، در زمان بوت سیستم کافیست برنامه مورد نظر را بدین طریق در crontab قرار دهیم:

@reboot /path/to/my/program @reboot updatedb

این برنامه در هر بار بوت مجدد سیستم اجرا خواهد شد. در ادامه مثالهایی برای درک بهتر موضوع آورده شده است :

<command> @reboot هنگام بوت سیستم اجرا می شود

<command> @yearly هر سال اجرا می شود[* 1 1 0 0]

<command> @annually هر سال اجرا می شود[* 1 1 0 0]

<command> @monthly هر ماه اجرا می شود[* * 1 0 0]

<command> @weekly هر هفته اجرا می شود[0 * * 0 0]

@daily <command> هر روز اجرا می شود[* * * 0 0]

@midnight <command> هر روز اجرا می شود[* * * 0 0

@hourly <command> هر ساعت اجرا می شود[* * * * 0]

برای اجرای چندین دستور پی درپی، آنها را با استفاده از "&&" به صورت پی در پی بنویسید. مثال زیر ابتدا دستور command_01 و سپس دستور command_02را در هر روز اجرا می کند :

```
@daily <command_01> && <command_02>
```

مثال های پیشرفته :

1) در مثال زیر در ساعت 3:12 دقیقه صبح هر روز از هر ماه، cron با اجرای این خط شروع به تهیه پشتیبان از /etc/ میکند. **dev/null 2>&1** به معنی ارسال هرگونه خروجی استاندارد به dev/null که سطل آشغال لینوکس است و هدایت خطاهای استاندارد 2 (standard error) به همان جایی که خروجی استاندارد رفته است بدون هر گونه خروجی در ترمینال یا هر نقطه دیگری از سیستم. اگر بجای >> از > استفاده کنیم در هر دفعه باز اجرا شدن دستور و فرستادن خروجی به فایل مربوطه , محتویات فایل پاک و خروجی جدید جایگزین میشود ولی با استفاده از >> خروجی به انتهای فایل افزوده خواهد شد.

```
12  3  *  *  *  root  tar cfz  /tohid/backup.tar.gz  /etc  >>  /dev/null  2>&1
```

2) در این مثال در تاریخ یکشنبه 7 oct ساعت 15:30 از ماه دهم روز 7 پیغام تبریکی برای کاربر ارسال میشود.

```
30  15  7  10  0  *  root  echo  "happy birthday,tohid!!"
```

3) مثال بالا را به شیوه زیر هم میتوان نوشت، دقت کنید حروف اول روز و یا ماه باید بزرگ نوشته شود:

```
30  15  7  Oct  Sun  *  root  echo  "happy birthday,tohid!!"
```

4) اگر می خواهید که کاربر tohid یک دستور را دقیقه 15 ، بعد از هر ساعت بدون توجه به تاریخ اجرا کند بدین طریق عمل میکنیم:

```
15 * * * * tohid  echo  "I'm skywan13 Remember"
```

5) یا اگر تمایل دارید هر 15 دقیقه اجرا بشود از این خط استفاده می کنیم :

```
*/15 * * * * tohid  echo "I'm skywan13 Remember"
```

6) برای اجرا شدن یک دستور هر 2 ساعت یعنی در 2 صبح, 4 صبح, 6 صبح و....12 ظهر, 2 بعد از ظهر, 4 بعدازظهر و.... از این خط بهره می بریم .

```
0 */2 * * * tohid  echo  "I'm skywan13 Remember"
```

7) با اقزودن ' , ' در یک فیلد امکان چندین مرتبه اجرا شدن دستور مورد نظر بدست میاد. مثلا برای اجرای دستور در ۱۵ و ۳۰ دقیقه گذشته از هر ساعت بدین طریق عمل میکنیم:

```
15,30  * * * *  tohid  echo "I'm skywan13 Remember"
```

8) برای اجرای دستور مورد نظر در یک زمان مشخص، برای اولین هفته ماهی که شما تمایل دارید در فیلد روز از ۱-۷ استفاده میکنیم (خط تیره به معنی تا و یا الی می باشد). در این خط مشخص کرده ایم در ۱۵ و ۳۰ دقیقه گذشته هر دو ساعت از روز 1 تا 7 این خط اجرا شود :

```
15,30  */2  1-7  *  * tohid  echo "I'm skywan13 Remember"
```

9) خط زیر هر 2 دقیقه به 2 دقیقه به صفحه ایندکس یک وب سرور وصل شده و آن را دانلود کرده و سپس در دایرکتوری کاربر ذخیره می کند:

```
*/2 * * * wget  http://192.168.1.10/index.php >>  /home/skywan13/cron
```

10) در ساعت 12:30 هر روز فایل های خالی دایرکتوری /tmp را پاک می کند. دستور find برای اجرا شدن توسط crond از مجوز های کاربر root استفاده می کند. باید ابتدا دستور crontab -e را اجرا کنید تا فایل crontab شما برای ویرایش باز شود.

```
30 0 * * * root find/tmp -type f -empty -delete
```

11) دستور زیر در 10 ژوئن ساعت 8:30 صبح یک backup توسط اسکریپتی گرفته می شود. توجه کنید برای ساعت 8:30 شب باید از 20:30 استفاده کنید.

```
30 8 10 06 * root /home/skywan13/full-backup
```

برای گرفتن backup در ساعت 11 ظهر و 4 بعد از ظهر(16) هر روز از دستور زیر استفاده کنید:

```
00 11,16 * * * /home/skywan13/bin/incremental-backup
```

12) برای انجام در روز های خاص از هفته مثل روز دوم هفته (یک شنبه روز اول هفته میلادی و عددش 0 است و دوشنبه روز دوم و عددش 1 است) تا روز ششم یعنی جمعه هر هفته بین ساعت های 9 صبح تا 6 عصر انجام می شود.

```
00 09-18 * * 1-5 /home/skywan13/bin/check-db-status
```

13) برای اجرای وظایف در یک محدوده زمانی خاص مثل بین ساعت 9 صبح تا 6 عصر (18) از دستور زیر استفاده می کنیم.

```
00 09-18 * * * /home/skywan13/bin/check-db-status
```

14) می خواهیم اسکریپتی به نام clean.cache که cache سیستم را پاک می کند، هر 10 روز یکبار اجرا شود. لذا فایل اسکریپت مربوطه را در مسیر /etc/cron.daily/ قرار می دهیم. محتوای اسکریپت به شرح زیر می باشد :

```
#!/bin/bash
# A sample shell script to clean cached file from lighttpd web server
CROOT=”/tmp/cachelighttpd/”
DAYS=10
LUSER=”lighttpd”
LGROUP=”lighttpd”
# start cleaning
/usr/bin/find ${CROOT} -type f -mtime +${DAYS} | xargs -r /bin/rm
```

```
# if directory deleted by some other script just get it back
if [ ! -d $CROOT ]
then
/bin/mkdir -p $CROOT
/bin/chown ${LUSER}:${LGROUP} ${CROOT}
fi
```

سپس برای اجرایی کردن اسکریپت از خطوط زیر استفاده می کنیم:

```
# crontab -l > /backup/cron/cronjobs.bakup
# crontab -u username -l > /backup/cron/cronjobs_username.bakup
```

اجرا برنامه ها با واسط گرافیکی کاربر

برنامه های که کاربر می خواهد اجرا کند به دو صورت می باشند :

برنامه هایی که دارای محیط گرافیکی کاربر هستند (GUI) و نیاز به تعامل با سرویس دهنده پنجره X هستند مانند مرورگرFirefox و برنامه هایی که بدون GUI هستند که این برنامه ها خروجی و ورودی آن ها در پوسته خط فرمان است و برای اجرا شدن نیازی به تعامل با سرویس دهنده پنجره X ندارند.

برنامه هایی که در دسته دوم (CLI) قرار می گیرند بدون هیچ مشکلی به وسیله Cron اجرا می گردند. اما برنامه های دسته اول که دارای GUI هستند فقط با نوشتن دستور مورد نظر اجرا نخواهند شد. قبل از دستور باید به سرویس دهنده X بگویید که برنامه در کدام صفحه نمایش برای شما اجرا شود.

برای این منظور قبل از دستور مورد نظر از env DISPLAY=:0 استفاده می کنیم. به عنوان مثال برای اجرای برنامه gedit در ساعت 10:25 هر روز صبح در فایل crontab خط زیر را وارد می کنیم :

```
25  10  *  *  *   env DISPLAY=:0   /usr/bin/gedit
```

DISPLAY=:0 به Cron می گوید که برنامه در صفحه جاری (desktop) اجرا شود.

و اگر دارای چندین صفحه نمایش هستید از دستور زیر استفاده می کنیم.

DISPLAY=:0.0 به Cron می گوید که برنامه در صفحه نمایش اول و در صفحه جاری اجرا شود. این دستور حتما باید قبل از اجرا شدن یک برنامه GUI به وسیله Cron اجرا شود. شما می توانید این دستور را در قسمت Startup Applications قرار دهید تا هنگام بوت شدن سیستم اجرا شود

```
25  10  *  *  *   env DISPLAY=:0.0   /usr/bin/gedit
```

زمانبندی اجرای فرامین توسط Anacron

anacron را میتوان برای اجرای دستورات بصورت دوره ای استفاده کرد که این فواصل به روز تعیین می شوند. برخلاف cron , تصور نمی کند که دستگاه 24 ساعت بطور مستمر روشن است. از این رو آن را میتوان در دستگاههایی که 24 ساعت روشن نیستند برای کنترل jobهای روزانه هفتگی و ماهانه که بطور معمول با cron صورت میگیرد استفاده کرد. بنابراین درanacron موضوع اصلی اجرا شدن job هاست نه اجرا شدن اونها سر ساعت و دقیقه تعیین شده همانند cron.

cron-daily در لینوکس CentOS در ساعت چهار و دو دقیقه اجرا می شود حال فرض کنید سیستم راس ساعت 4 خاموش شود و در ساعت 5 مجددا UP شود و این مدت 23 ساعتی که باید به چهار و دو دقیقه بعدی برسد ممکن است در سیستم کلی اتفاق رخ دهد که ما در این صورت یک روز کامل را از دست داده ایم. یا فرض کنید سیستمی داریم که باید 5 روز به 5 روز از آن بکاپ بگیریم. اگر 5 روز اول را از دست بدهیم و بکاپ دوم را با cron بگیریم در اصل ده روز را از دست داده ایم. این خلاء یک نقص برای سیستم و سرویس cron محسوب می شود. این ضعف با سرویسی به نام anacron پوشش داده می شود.

کار anacron اجرای cron ای است که از دست رفته و زمان اجرای آن گذشته و اجرا نشده است. anacron برای سیستم های است که UP و DOWN زیادی دارند مثل سیستمهای خانگی و یا کلاینتهای تحت شبکه. محدودیت cron به دقیقه بود اما anacron محدود به روز است. یعنی اگر cron ساعتی یک بار کاری را انجام ندهد anacron یک روز بعد شروع به انجام آن می کند. این سرویس مناسب سرور نیست چون محدودیت زمانی آن به روز می باشد.

برای استفاده از anacron می بایست بسته ی مربوط نصب و سپس سرویس anacron را اجرا کنیم.

```
# yum –y install cronie-anacron
# rpm  -qa  |  grep   cronie-anacron
# rpm  -qi cronie-anacron
```

تنظیم کردن وظایف Anacron

(tasks)وظایف anacron در فایل etc/anacrontab/ لیست شده است. علاوه بر این ، در دایرکتوری var/spool/anacron/ هم برای خودش مانند cron یک دایرکتوری مستقل دارد.

محتویات فایل etc/anacron/ تقریبا شبیه فایل کانفیگ cron است اما تفاوتهایی هم دارد. خطوط اصلی و متفاوت این فایل در انتهای آن قرار دارد که حاوی 4 فیلد اصلی می باشد. هر خط در این فایل تنظیمات مربوط به یک وظیفه است و بدین ترتیب نوشته شده اند :

Period	Delay	Job-identifier	Command	
1	65	cron.daly	ran-parts	/etc/……

period :

عدد روزهایی که باید بین اجرای دستورات طی بشود مثلا 9 ، یعنی دستور هر 9 روز یکبار اجرا میشود یا 7 برای اجرای هفتگی است. این فیلد بر مبنای روز می باشد.

delay :

برای هر job , anacron بررسی میکند که آیا این دستور در n(n=period) روز گذشته اجرا شده است یا نه. اگر نشده باشد anacron آن را اجرا میکند. در اصل این فیلد زمانی است که سیستم بعد از UP شدن شروع به انجام کار عقب افتاده می کند. مبنای زمانی این فیلد دقیقه می باشد.

Job-identifer :

آخرین زمانی که anacron یک کاری را انجام داده است در این فایل ثبت می شود.

command :

دستوراتی که خواهان اجرای آن هستیم.

خطوط زیر در فایل اصلی کانفیگ anacron آمده که به ترتیب توضیح داده می شود.

```
1   65   cron.daily
7   70   cron.weekly
30   75   cron.monthly
```

cron.daily 65 1: این خط می گوید یک روز به یک روز که سیستم up می شود 65 دقیقه صبر کند سپس این فایل را بررسی کند، اگر روز قبل این کار انجام شده باشد که هیچ، در غیر این صورت باید job مورد نظر انجام شود.

cron.weekly 70 7: این خط بیان می کند هر 7 روز به 7 روز که سیستم up شد 70 دقیقه صبر کندو job مورد نظر را در صورت انجام نشدن انجام دهد.

cron.monthly 75 35: این خط هم می گوید هر 30 روز یکبار که سیستم بالاآمد 75 دقیقه صبرکند بعد job مورد نظر را انجام می دهد حال فرض کنید 15 روز از اول ماه گذشته و سیستم را up می شود anacron توسط این خط بررسی می کند چون 15 روز به سر ماه بعدی مانده ، job مربوطه را اجرا نمی کند.

متغیرهای محیطی همچون SHELL و PATH را در بالای فایل /etc/anacrontab میتوانید تنظیم کنید چنانکه در cron هم تنظیم میکردیم.

برای period میتوانید هم از اعداد برای نشان دادن دوره ی اجرای دستور استفاده کنید هم از نشانه هایی مانند زیر :

@daily
@weekly
@monthly

در این مثال هر روز با تاخیر یک دقیقه ای جمله ی hi,how are u today به انتهای فایل /skywan13/hi افزوده میشود.

@daily 1 skywan13 echo "hi,how are u today?" >> /skywan13/hi

نکته : این تایمی که بر اساس ساعت آمده به این علت است که سیستم بعد از up شدن به یک پایداری برسد.

نکته مهم : این سرویس به صورت پیش فرض stop است و اگر فعال شود هر روز علاوه بر کارهای خودش وظایف cron را هم انجام می دهد، آنوقت است که بین cron و anacron تضاد کاری پیش می آید. اگر anacron زودتر از cron اجرا شود، cron متوجه نمی شود و هر دو job مورد نظر را انجام می دهند.

وقتی anacron یک job را برای دفعه اول اجرا میکند فایلی همنام با job-identifer را در /var/spool/anacron/ میسازد که محتوای فایل تاریخ اجرای job است (نه ساعت).اصطلاحا به این فایل timestamp گفته میشود. بعد از اجرای مجدد این job دوباره با تاریخ بازنویسی می شود.

کاربر عادی در حالت پیش فرض قادر به استفاده از anacron نیست به یک دلیل ساده ، چونکه اجازه ساخت فایل timestamp را در دایرکتوری var/spool/anacron/ ندارد.

برای حل این مشکل بدون اینکه مشکل جدیدتری بوجود آید بدین ترتیب عمل میکنیم :

1- یک گروه میسازیم و کاربرها را به آن اضافه میکنیم:

برای انجام اینکار از groupadd یا addgroup میتوانید استفاده کنید :

```
# groupadd  anacron
or
# addgroup  anacron
```

حالا شما یک گروه بدون کاربر دارین که باید کاربران مورد نظرتون را به این گروه اضافه نمایید:

```
#adduser  skywan13  anacron
```

2- مجوز مالکیت /var/spool/anacron را تغییر میدهیم برای تغییر مالکیت حتما باید با کاربر root وارد شده باشید:

```
# chown  root.anacron  /var/spool/anacron
# chmod  g+w  /var/spool/anacron
```

خوب حالا شما عضو گروه anacron هستید و مجوز نوشتن را در دایرکتوری مربوطه دارید.

3- برای ادامه کار باید فایل anacron مربوط به خودتان را ایجادکنید:

فایل anacrontab را به یک جایی در دایرکتوری خانه خودتان مثلا /home/skywan13/anacron کپی کنید و طبق آموزش های بالا فایل را تنظیم نمائید.

4- anacron یک daemon نیست و فقط هنگام بالا آمدن سیستم برای کاربر root اجرا میشود پس باید برای خودتان هم اجرا کنید. بدین ترتیب عمل میکنیم:

```
# echo anacron -t $HOME/etc/anacrontab >> .bashrc

# echo anacron -t $HOME/etc/anacrontab >> .bash_profile
```

زمان‌بندی دستورات با at

خیلی وقت ها شده که بخواهیم یک دستور را برای اجرا در زمانی خاص زمان‌بندی کنیم. مثلا ممکن است در ساعاتی از شب دریافت فایل از اینترنت رایگان باشد ولی در آن ساعات خواب باشیم و بیدار ماندن سخت! برای حل این مشکل در ویندوز IDM داشتیم، در لینوکس چه کنیم ؟

در این جا نرم‌افزاری رو به شما معرفی می کنم به نام at که برای برنامه‌ریزی کردن دستورات است و کار با آن نیز بسیار ساده است. at تقریبا در همه توزیع‌های لینوکس نصب است. at فایل یا اسکریپت را اجرا نمی کند بلکه فقط توانایی اجرای یک دستور، در یک زمان خاص را دارد و البته دوره زمانی هم ندارد. خروجی دستور at به یوزر استفاده کننده میل می شود و با ریست سیستم هم خط نوشته شده at از بین میرود.

نحوه نوشتن خط at :

```
  at    time    date
# at    now     +1min
at> ls /etc
```

یک مثال ساده

دستورات زیر را در ترمینال وارد کنید:

```
# at 2:00
at> echo \"Hello World!\" >> /home/$USER/log
```

و سپس Ctrl + D بزنید.

مثال بالا از at می‌خواهد که دستور خط دوم را که خود عبارت Hello World! را در لاگ کاربر کپی کند و در ساعت ۲ صبح اجرا کند. زدن Ctrl + D بعد از وارد کردن خط دوم، به at پایان وارد کردن لیست کارها را اعلام می‌کند.

برای این‌که at بتواند دستورات را اجرا کند باید Daemon آن در حال اجرا باشد:

```
# service  atd  start
```

همان‌جور که مشخص است با at می‌توان تمامی کارها را زمان‌بندی کرد. فرض کنید لیستی از فایل‌ها برای دانلود دارید و می‌خواهید دانلود ساعت ۲ صبح شروع شده و در ساعت ۸ صبح پایان یابد. ابتدا لینک‌های مورد نظر را در فایلی متنی به طوری که هر لینک در یک خط باشد کپی کنید.

در این مثلا ما فایل را dl-list.txt نامیدیم و از نرم‌افزار دانلود Aria2 برای دانلود کمک گرفتیم:

```
# at 2:00  +  1000  days
at> aria2c  -i  ~/dl-list.txt  -j  1 -x  5
```

و سپس Ctrl + D مثال بالا فرمان دانلود را با کمکat ، به مدت ۱۰۰۰ روز پیاپی در ساعت ۲ صبح اجرا می‌کند.

حالا برای بسته شدن دانلود در ۸ صبح:

```
# at  8:00  +  1000  days
at>  pkill  aria2c
```

بعضی از وب‌سایت‌ها ممکن است فایل را تنها در اختیار کاربرانی که در آن وب‌سایت حساب دارند بگذارند مثل Rapidshare که در آن صورت کافیست نام کاربری و رمزعبور خود را در قالب اطلاعات درخواست دانلود با aria2 بفرستید:

```
at> aria2c -i ~/dl-list.txt -j 1 -x 5 --http-user=ali --http-passwd=123456
```

دستور بالا فایل‌های لیست شده در dl-list.txt را با نام کاربری ali و رمزعبور 123456 دانلود می‌کند.

جزییات at

atq لیست دستورات برنامه‌ریزی شده را به همراه شماره آن‌ها نشان میدهد.

atrm دستورات برنامه‌ریزی شده را پاک می کند:

```
# atrm que_id
```

برای یافتن que_id دستور مورد نظر، از atq کمک بگیرید.

قالب زمانی به صورت HH:MM وارد می‌شود، استفاده از am و pm هم معتبر است. مثلا ۸ صبح در مثال بالا را 8 am هم می‌توان نوشت.

تاریخ به صورت [CC]YY-MM-DD باید وارد شود، از مخفف ماه و روزها نیز می‌توان استفاده کرد.

عبارت‌هایی مثل فردا، امروز، عصر و نیمه شب هم معتبر هستند.

```
sun mon tue wed thu fri sat
jan feb mar apr may jun jul aug sep oct nov dec
tomorrow today noon midnight
```

برای تکرار یک کار در چند روز:

```
+ N days
```

چند زمان‌بندی پیچیده‌تر با at :

```
at 3:00pm  tomorrow
at 2:00am  jul  5 + 4  days
at 2:00  2012-7-5
at 2:00  wed
```

نکته:

واحد های زمانی کوچک تر در اول قرار دارند. یعنی مثلا ساعت و دقیقه قبل از ماه.

aria2فقط یک نمونه برنامه برای دانلود است؛ Axel, wget و lftp از دیگر مثال‌ها هستند.

atd همان طور که در بالا گفته شد یکی از فرمان ها at است . برای استفاده از آن میشود از سیلابس هایی جهت نشان دادن دقیق زمان استفاده کرد که به آن ها میپردازیم. همانطور که در قبل اشاره شد از دستور at برای کارهایی که یکبار انجام میشوند استفاده میشود:

```
at now
at 04:11 am
at now +5 min
at now +5 hours
at now +4 days
at now +4 weeks
at 13:13 pm October 18
```

برای مثال با این فرمان به این صورت میتوان در ۵ دقیقه آینده سیستم را خاموش کرد:

```
# at now +5 min
at> date > /file1
```

پس از اعلان سیستم از شما اطلاعات مربوط را میگیرد و در زمان تعیین شده کار را انجام میدهد . برای مشاهده لیست کار هایی که توسط این فرمان صورت خواهد کرفت از فرمان at -l ویا atq استفاده میکنیم .

```
# at  -l
2   Sat Aug 24 10:35:00 2013 a Ali
# atq
2    Sat Aug 24 10:35:00 2013 a Ali
```

منابع :

مطالب متفرقه منتشر شده در اینترنت

سر فصل دوره های RHCE و LPIc2

راهنما cron & anacron & at

سایت centos .org