جامعة البعث

الكلية التطبيقية-السنة الثالثة -تقنيات حاسوب

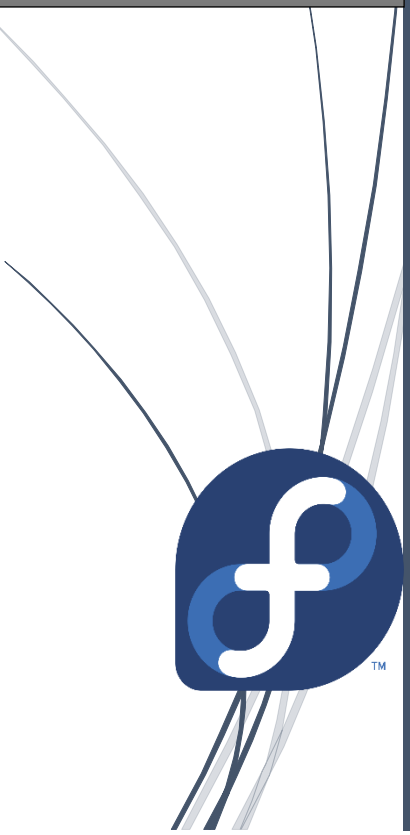
الفصل الأول – القسم العملي

Network Operating Systems 1

Lecture5

2022-2021

Supervised by



إدارة المستخدمين والمجموعات وإدارة المهام والإجرائيات

ملفات إدارة المستخدمين والمجموعات (Users and Group files)

تستخدم Fedora هيكلية تحدد مجموعة خاصة للمستخدم user private group (UPG)، حيث يتم إنشاء مجموعة خاصة بالمستخدم كلما تمت إضافة مستخدم جديد إلى النظام. تمتلك هذه المجموعة نفس اسم المستخدم الذي تم إنشاؤها من أجله وهذا المستخدم هو العضو الوحيد في المجموعة user private group.

يتم تعيين أذونات افتراضية لملف أو دليل تم إنشاؤه حديثًا من قبل مستخدم، مما يسمح لكل من المستخدم ومجموعة هذا المستخدم (UPG) بإجراء تعديلات على الملف أو الدليل.

إنّ كل ملفات إدارة المستخدمين والمجموعات توجد ضمن **/etc/** وسنستعرضها فيما يلي:

1. **ملف passwd:** وهو ملف يحوي معلومات عن كل المستخدمين الموجودين في النظام حيث تكون بياناته وفق الشكل التالي:

Username:x:UID:GID:comment:homedirectory:shell

حيث تدل ال comment على تعليق معين يُمثّل الاسم الكامل للمستخدم.

```
ali:x:1000:1000:Ali Mustafa:/home/ali:/bin/bash
alim:x:1001:1001:Ali_Mustafa:/home/alim:/bin/bash
alimustafa2:x:1002:1002:ali mustafa2:/home/alimustafa2:/bin/bash
```

2. **ملف shadow:** وهو ملف يحوي كلمات مرور المستخدمين ولكن مشفّرة حيث تكون بياناته وفق الشكل التالي:

Username:'encrypted password'

```
ali:$6$EHUkBulCDEDlmlA3
$2fFhb6QKPKO0ureL9d3EBVz2LdbbJfC.uUbCkk6ZaoAYtdT0ubHSocwHjPvj2dlAhtqNlct43Y4FvyApMO49A.:18562:0:99999:7:::
alim:$6$SJtDB9QZ$.GMaWVL4iBP/
lRX9q86EW9Hy.hldsRXM4wb.b2SeeV6TcjTgh2BDFreXD1Q2tuLPEmx3AUzRL.xqf50jwsYT11:18577:0:99999:7:::
alimustafa2:$6$MZ1mYj1oizEuTszV$9utLP29H3kx.H2MVJzAuYtpbO4./
iRxBGG3VN0031eaRbvYsRRTi/2dRZNuCYgJdnMi5sCVKJP0wIx4Kvm4iX0:18577:0:99999:7:::
```

يمكن أن تكون كلمة المرور مسبوقة بإشارة تعجّب (!) أي أن الحساب مُقفل، وإذا وُجِدت إشارة التعجّب دون كلمة المرور المُشفّرة بعدها فذلك يعني أن الحساب مُقفل ولم يستخدم المستخدم هذا الحساب بعد ولم يضع له كلمة مرور.

3. **ملف group:** يتم فيه تخزين قائمة بجميع المجموعات مع معرّف فريد لكل مجموعة، ويحوي كل سطر معلومات كالتالي:

Group:x:GID:

```
ali:x:1000:
alim:x:1001:
alimustafa2:x:1002:
```

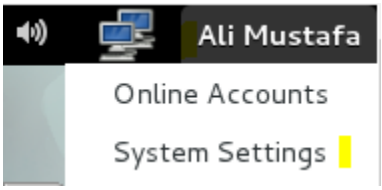
4. **ملف Gshadow:** يحوي نفس معلومات الملف group ولكن بدل x يتم وضع كلمة المرور مُشفّرة

```
ali:!!::
alim:!::
alimustafa2:!::
```

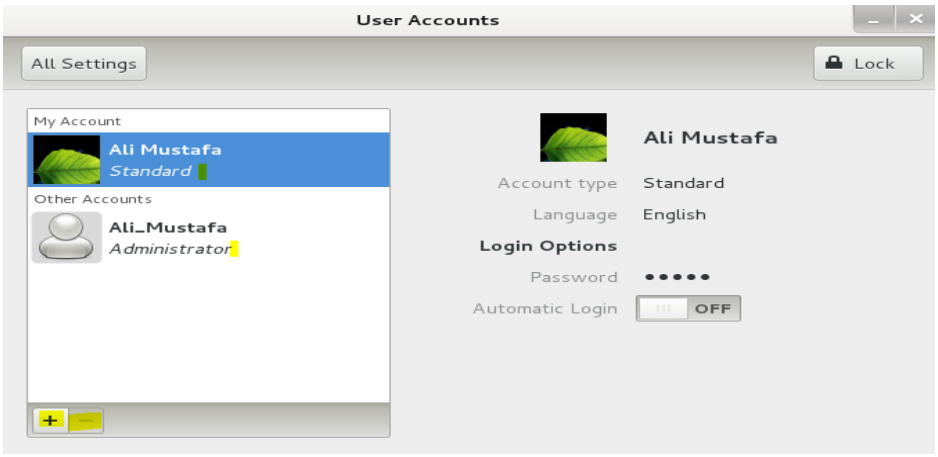
إضافة وحذف وتعديل المستخدمين والمجموعات

1-إدارة المستخدمين باستخدام واجهة المستخدم الرسوميّة

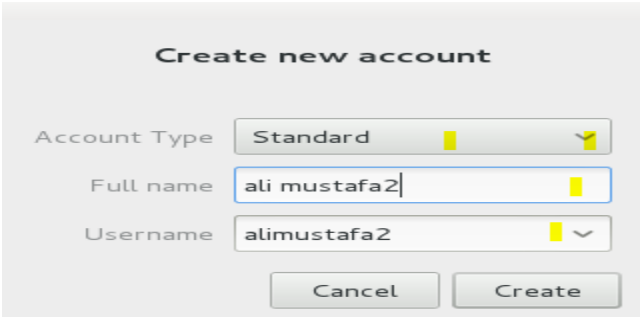
ندخل من اسم المستخدم بالأعلى إلى إعدادات النظام:



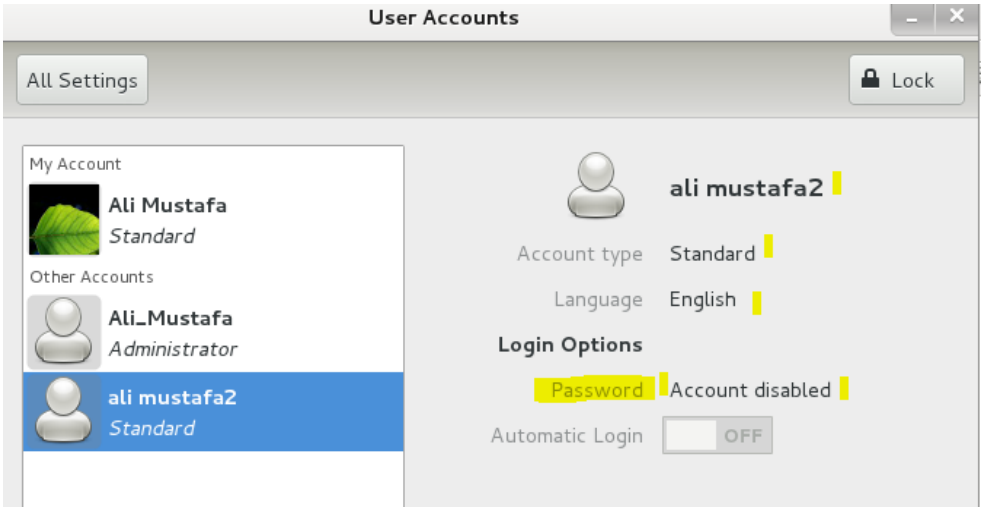
ندخل بعدها إلى User Accounts , ويمكننا إضافة أو حذف مستخدم باستخدام إشارتي (+) و (-):



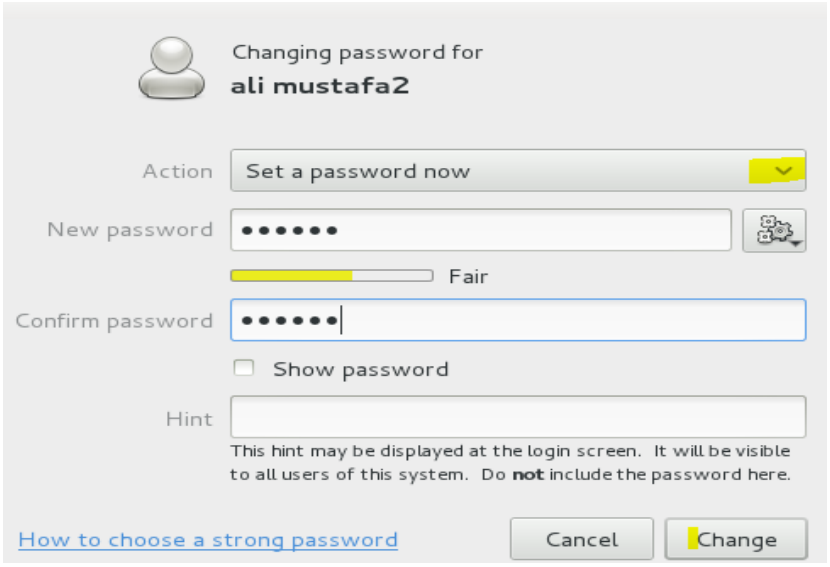
نضغط على (+) لإضافة حساب مستخدم جديد ويمكننا بعدها تحديد نوع الحساب فيما إذا كان عادي (Standard) أو إداريّ (Administrator) مع تحديد الاسم الكامل مع اسم المستخدم كالتالي:



نلاحظ أنّه وبعد إضافة المستخدم سيكون حسابه غير مُفعّل ويجب تفعيله وذلك بتعيين كلمة مرور له:



حيث نضغط على Account disabled فتظهر الواجهة التالية لتعيين كلمة المرور وطريقة تحديدها:



وبذلك يصبح الحساب مُفعّلاً وجاهزاً للاستخدام وتسجيل الدخول عبره.

2-إدارة المستخدمين والمجموعات باستخدام أدوات سطر الأوامر (managing users and groups using command line utilities)

نوضّح بالجدول التالي أدوات بالCommand Line لإدارة المستخدمين والمجموعات:

Utilities	Description
Id	**Displays user and group IDs.**
Groups	**Displays the groups that the user belong.**
useradd, usermod, userdel	**Standard utilities for adding, modifying, and deleting user accounts.**
groupadd, groupmod, groupdel	**Standard utilities for adding, modifying, and deleting groups.**
gpasswd	**Standard utility for administering the /etc/group configuration file.**
pwck, grpck	**Utilities that can be used for verification of the password, group, and associated shadow files.**
pwconv, pwunconv	**Utilities that can be used for the conversion of passwords to shadow passwords, or back from shadow passwords to standard passwords.**
grpconv, grpunconv	**Similar to the previous, these utilities can be used for conversion of shadowed information for group accounts.**

إضافة مستخدم جديد

لإضافة مستخدم جديد للنظام، نكتب ما يلي على سطر الأوامر (مع الدخول كمستخدم إداريّ root)

```
useradd options username
```

افتراضيّاً، يقوم الأمر **useradd** بإنشاء حساب مستخدم مُقفل (Locked user account)، ويجب فك قفله باستخدام الأمر التالي (أيضاً يجب الدخول كمستخدم إداريّ root):

```
passwd username
```

ونستعرض ضمن الجدول التالي أبرز الخيارات التي تُستخدم مع الأمر useradd:

Option	Description
-c '*comment*'	***comment* can be replaced with any string. This option is generally used to specify the full name of a user.**
-d *home_directory*	**Home directory to be used instead of default /home/*username*/.**
-e *date*	**Date for the account to be disabled in the format YYYY-MM-DD.**
-f *days*	**Number of days after the password expires until the account is disabled. If 0 is specified, the account is disabled immediately after the password expires. If -1 is specified, the account is not disabled after the password expires.**

-g *group_name*	**Group name or group number for the user's default (primary) group. The group must exist prior to being specified here.**
-G *group_list*	**List of additional (supplementary, other than default) group names or group numbers, separated by commas, of which the user is a member. The groups must exist prior to being specified here.**
-m	**Create the home directory if it does not exist.**
-M	**Do not create the home directory.**
-N	**Do not create a user private group for the user.**
-p *password*	**The password encrypted with crypt.**
-r	**Create a system account with a UID less than 1000 and without a home directory.**
-s	**User's login shell, which defaults to /bin/bash.**
-u *uid*	**User ID for the user, which must be unique and greater than 999.**

مثال:

```
[root@Ali ~]# useradd -c 'alimustafa3' ali3
[root@Ali ~]#
[root@Ali ~]# passwd ali3
Changing password for user ali3.
New password:
BAD PASSWORD: it is too simplistic/systematic
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
```

تعديل معلومات مستخدم

يتم باستخدام الأمر **usermod** الذي يقبل نفس الخيارات الموضّحة بالجدول السابق.

مثال: يمكن إلحاق المستخدم بمجموعة أُخرى مختلفة عن مجموعته الأساسية كالتالي:

```
[root@Ali ~]# usermod -aG root ali3
```

حيث نستخدم الخيار **a**– لإلحاق (append) مستخدم (ali3 في مثالنا) بمجموعة ثانوية (ولتكن root) نُحددها بواسطة الخيار **G**–.

نبدّل للمستخدم الذي أجرينا لتعديل عليه ونتحقق من المجموعات التي ينتمي إليها كالتالي:

```
[root@Ali ~]# su ali3
[ali3@Ali root]$ groups
ali3 root
```

في حال لم يتم استخدام الخيار a– مع الخيار G–، سيتم تغيير المجموعة التي ينتمي إليها المستخدم إلى المجموعة التي تم تحديدها مع الخيار G– فقط ولن يتم إضافة المستخدم لمجموعة مختلفة عنها (أي سيتم تجاهل أي مجموعة كان ينتمي إليها المستخدم وسبق تحديدها عبر الخيار G-).

```
[root@Ali ~]# usermod -G ali ali3
[root@Ali ~]# su ali3
[ali3@Ali root]$ groups
ali3 ali
```

نلاحظ أن المستخدم أصبح ينتمي لمجموعته الافتراضية وللمجموعة ali دون المجموعة السابقة root.

حذف مستخدم

نستخدم الأمر **userdel** لحذف حساب مستخدم بشرط أن يكون مسجّلاً للخروج (logged out).

```
[root@Ali ~]# userdel ali4
```

```
[root@Ali ~]# su ali4
su: user ali4 does not exist
```

إضافة مجموعة جديدة

نستخدم الأمر groupadd لإضافة مجموعة جديدة للنظام , ونكتب كالتالي بعد الدخول كمستخدم root:

```
groupadd options group_name
```

ونوضّح بالجدول التالي الخيارات الممكن استخدامها مع الأمر groupadd:

Option	Description
-f, --force	When used with -g *gid* and *gid* already exists, groupadd will choose another unique *gid* for the group.
-g *gid*	Group ID for the group, which must be unique and greater than 999.
-o, --non-unique	Allows creating groups with duplicate GID.
-p, --password *password*	Use this encrypted password for the new group.
-r	Create a system group with a GID less than 1000.

مثال 1: لننشئ المجموعة students ذات المُعرّف 1500:

```
[root@Ali ~]# groupadd -g 1500 students
```

مثال 2: إنشاء مجموعة إداريّة (System group) ولتكن Admins مع تحديد كلمة مرور لها:

```
[root@Ali ~]# groupadd -r -p 123456 Admins
```

الانتقال بين المجموعات

نستخدم الأمر **newgrp**:

```
[root@Ali ~]# newgrp Admins
[root@Ali ~]# id
uid=0(root) gid=983(Admins) groups=0(root),983(Admins)
nfined_r:unconfined_t:s0-s0:c0.c1023
[root@Ali ~]# newgrp root
[root@Ali ~]# id
uid=0(root) gid=0(root) groups=0(root),983(Admins) cont
ed_r:unconfined_t:s0-s0:c0.c1023
```

حذف مجموعة

يتم باستخدام الأمر **groupdel**:

```
[root@Ali ~]# groupdel students
[root@Ali ~]#
[root@Ali ~]# newgrp students
newgrp: group 'students' does not exist
```

أمن الحساب (Account security)

فيما يلي بعض الطرق المتّبعة لأمن وحماية حساب المستخدمين:

1) **تفعيل مدة محدّدة لكلمة المرور (Enabling Password Aging)**
يتم ذلك باستخدام الأمر **chage** كالتالي:

```
chage options username
```

وفيما يلي جدول يوضّح الخيارات المتاحة مع الأمر chage:

Option	Description
-d *days*, --lastday	**Specifies the number of days since January 1, 1970 the password was changed. The date may also be expressed in the format YYYY-MM-DD.**
-E *date*, --expiredate	**Specifies the date on which the account is locked, in the format YYYY-MM-DD. A user whose account is locked must contact the system administrator before being able to use the system again.**
-I *days*, --inactive	**Specifies the number of inactive days after the password expiration before locking the account. If the value is 0, the account is not locked after the password expires.**
-l, --list	**Lists current account aging settings.**
-m *days* , --mindays	**Specify the minimum number of days after which the user must change passwords. If the value is 0, the password does not expire.**
-M *days*, --maxdays	**Specify the maximum number of days for which the password is valid. When the number of days specified by this option plus the number of days specified**

	with the -d option is less than the current day, the user must change passwords before using the account.
-W *days* , --warndays	**Specifies the number of days before the password expiration date to warn the user.**

مثال 1:

عرض قيم معلومات password Aging الافتراضية لحساب معيّن:

```
[root@Ali ~]# chage -l ali3
Last password change                                    : Nov 30, 2020
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

مثال 2:

تعيين يوم محدّد لانتهاء صلاحية كلمة المرور للمستخدم ويجب تغييرها عند هذا التاريخ:

```
[root@Ali ~]# chage -d 2020-11-30 ali3
```

- عندما يتبع الأمر chage مباشرةً باسم المستخدم (أي عندما لا يتم تحديد خيارات سطر أوامر)، فإنه يعرض قيم ال aging لكلمة المرور الحالية للمستخدمين المحددين ويسمح لك بتغيير هذه القيم بشكل تفاعلي.

```
[root@Ali ~]# chage ali3
Changing the aging information for ali3
Enter the new value, or press ENTER for the default

        Minimum Password Age [0]: 30
        Maximum Password Age [99999]: 60
        Last Password Change (YYYY-MM-DD) [2020-11-30]:
        Password Expiration Warning [7]:
        Password Inactive [-1]:
        Account Expiration Date (YYYY-MM-DD) [1969-12-31]:

[root@Ali ~]# chage -l ali3
Last password change                                    : Nov 30, 2020
Password expires                                        : Jan 29, 2021
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 30
Maximum number of days between password change          : 60
Number of days of warning before password expires       : 7
```

يمكنك إعداد كلمة المرور بحيث تنتهي صلاحيتها في المرة الأولى التي يسجل فيها المستخدم الدخول. وهذا يفرض على المستخدمين تغيير كلمات المرور على الفور.

```
[root@Ali ~]# chage -d 0 ali3
```

وسيُطلب بعدها من المستخدم إدخال كلمة مرور جديدة.

2) تفعيل تسجيل الخروج التلقائيّ (Enabling Automatic Logouts)

خاصةً عندما يتم تسجيل دخول المستخدم ك root، قد تُشكّل جلسة تسجيل الدخول غير المراقبة مخاطرة أمنية كبيرة. لتقليل هذه المخاطر، يمكننا إعداد النظام لتسجيل خروج المستخدمين الغير نشطين تلقائيًا بعد فترة زمنية محددة:

1- نتأكد أولاً من أن الحزمة **screen** مُنصّبة وجاهزة للاستعمال:

```
[root@Ali ~]# yum install screen
```

2- ندخل كمستخدم root ونضيف السكر التالي إلى بداية الملف /etc/profile للتأكد من عدم مقاطعة معالجة هذا الملف:

```
trap "" 1 2 3 15
```

3- نضيف الأسطر التالية إلى نهاية الملف /etc/profile لبدء جلسة screen session في كل مرة يسجّل بها المستخدم الدخول:

```
SCREENEXEC="screen"
if [ -w $(tty) ]; then
  trap "exec $SCREENEXEC" 1 2 3 15
  echo -n 'Starting session in 10 seconds'
  sleep 10
  exec $SCREENEXEC
fi
```

لاحظ أنه في كل مرة تبدأ جلسة جديدة، سيتم عرض رسالة وسيتعيّن على المستخدم الانتظار عشر ثوانٍ.

4- أضف الأسطر التالية إلى ملف الإعدادات /etc / screenrc لإغلاق جلسة الشاشة (screen session) بعد فترة معينة من عدم النشاط:

```
idle 120 quit
autodetach off
```

سيؤدي هذا إلى ضبط الحد الزمني على 120 **ثانية**.

يمكنك بدلاً من ذلك إعداد النظام لقفل الجلسة فقط باستخدام الأسطر التالية:

```
idle 120 lockscreen
autodetach off
```

بهذه الطريقة، سيتم طلب كلمة المرور لفتح الجلسة مرة أُخرى.

تسري التغييرات السابقة في المرة المُقبلة التي يسجل فيها المستخدم الدخول إلى النظام.

إدارة المهام والإجرائيات (Process Administration)

من أجل إدارة نظام Fedora بشكل فعّال، يجب أن تكون قادرًا على مراقبة الإجرائيات والتحكّم فيها.

توفّر Fedora أدوات متعددة لمراقبة نشاط الإجرائية واستخدام الموارد وتعديل أولوية الإجرائية وإنهاء الإجرائيات.

معالجة الإجرائيات (Processes and Zombies)

- الإجرائية (Process) هي نسخة قيد التشغيل من البرنامج وإذا قمت بتشغيل برنامج مرتين سيتم إنشاء إجرائيتين.

يتم تحديد الإجرائيات برقم معرّف الإجرائية (PID)، والذي يتم تعيينهُ بشكل تسلسلي وهناك مجموعة من المعلومات المرتبطة بكل إجرائية، تتضمن ما يلي:

- **nice:**
قيمة تُستخدم لتغيير أولوية جدولة الإجرائية ، والتي تحدّد مقدار وقت المعالجة التي تتلقاها الإجرائية بالإضافة إلى مقدار وقت المعالجة. هذه القيمة موروثة من قبل الإجرائيات الأبناء (child processes) التابعة للإجرائية الأب (Parent process).
- **Parent process ID:**
معرّف الإجرائية الأب التي كانت السبب في بدء الإجرائية الابن.
- **real user ID and effective user ID:**
المعرّف العددي للمستخدم والمستخدم الذي يشغّل الإجرائية.
- **real group ID and effective group ID:**
المعرّف العددي للمجموعة والمجموعة التي تشغّل الإجرائية.
- **umask:**
السماحيات التي تمت وراثتها من الإجرائية الأب.

من المهم أن ندرك أنه في أي وقت معيّن تكون معظم الإجرائيات في حالة توقّف (Sleeping) أثناء انتظار توفّر بعض الموارد. قد يكون هذا المورد عبارة عن نقرة بالفأرة أو ضغطة مفتاح أو رزمة من الشبكة أو بعض البيانات من القرص أو انتظار وقت معيّن من اليوم.

- Zombie Processes

بمجرد انتهاء الإجرائية، يجب إزالتها من جدول الإجرائيات (Processes Table) حيث يمكننا رؤية الإجرائيات قيد التشغيل حالياً ضمن مراقب النظام "System-Monitor".

يتم استبدال المعرّف pid للإجرائية الابن بالمعرّف pid للإجرائية الأب حتى تستلم الإجرائية الأب التنفيذ و تقوم بإزالة جميع الإجرائيات الأبناء التي ماتت أو اكتمل تنفيذها.

Imagine it like this:
"You find a dead body in the middle of the road, you call the dead body's family and they take that body away from the road."

لكن الكثير من البرامج ليست مبرمجة جيدًا بما يكفي لإزالة الأبناء الميتة (child zombies) لذا فإن الشيء الوحيد المضمون لإزالة Child Zombies هو قتل الإجرائية الأب.

لكن في بعض الأحيان تظل بعض هذه الإجرائيات في جدول الإجرائيات حتى بعد إتمامها للتنفيذ.

لذا فإن هذه الإجرائيات التي أكملت تنفيذها ولكنها لا تزال موجودة في جدول الإجرائيات تسمى "**Zombie Processes**".

لا يوجد أي ضرر على نظام التشغيل عند وجود zombie process نظرًا لأنها لا تقوم بأي عمل، ولا تستخدم أي موارد أو تؤثر على أي إجرائية أُخرى.

ولكن، إذا كان يوجد الكثير من zombie processes، فسوف يزدحم جدول الإجرائيات وقد يصبح الأمر صعبًا بالنسبة للإجرائيات التي تعمل بالفعل.

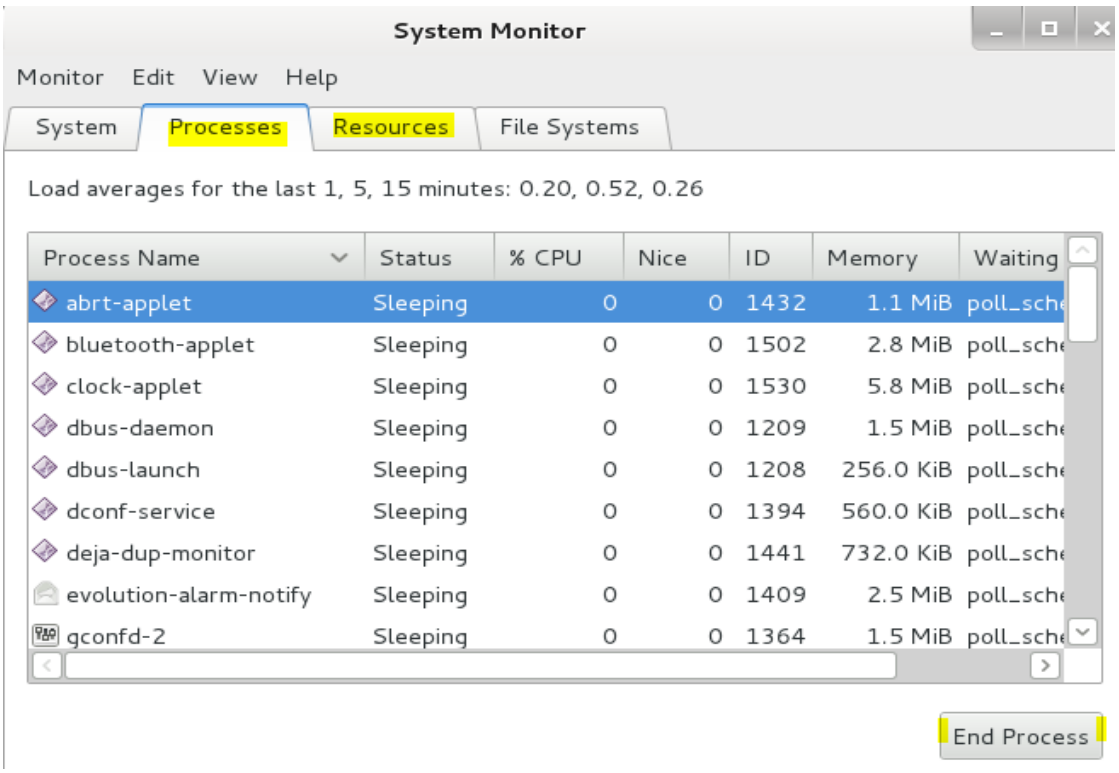
بكتابة الأمر **ps aux | grep Z** في الterminal سنحصل على تفاصيل جميع zombie processes في جدول الإجرائيات.

نكتب الأمر التالي لاستبدال معرّف الإجرائية بمعرّف الإجرائية الأب بحيث تُزيل الإجرائية الأب جميع الإجرائيات الأبناء التي ماتت أو اكتملت:

kill -s SIGCHLD *process_id*

مراقبة الإجرائيات بشكل رسوميّ في بيئة سطح المكتب Gnome

من القائمة Applications نختار System Tools ثمّ System Monitor وسيؤدي هذا لتشغيل برنامج **gnome-system-monitor** وتقديم العرض الموضح في الشكل التالي:



ملاحظة:

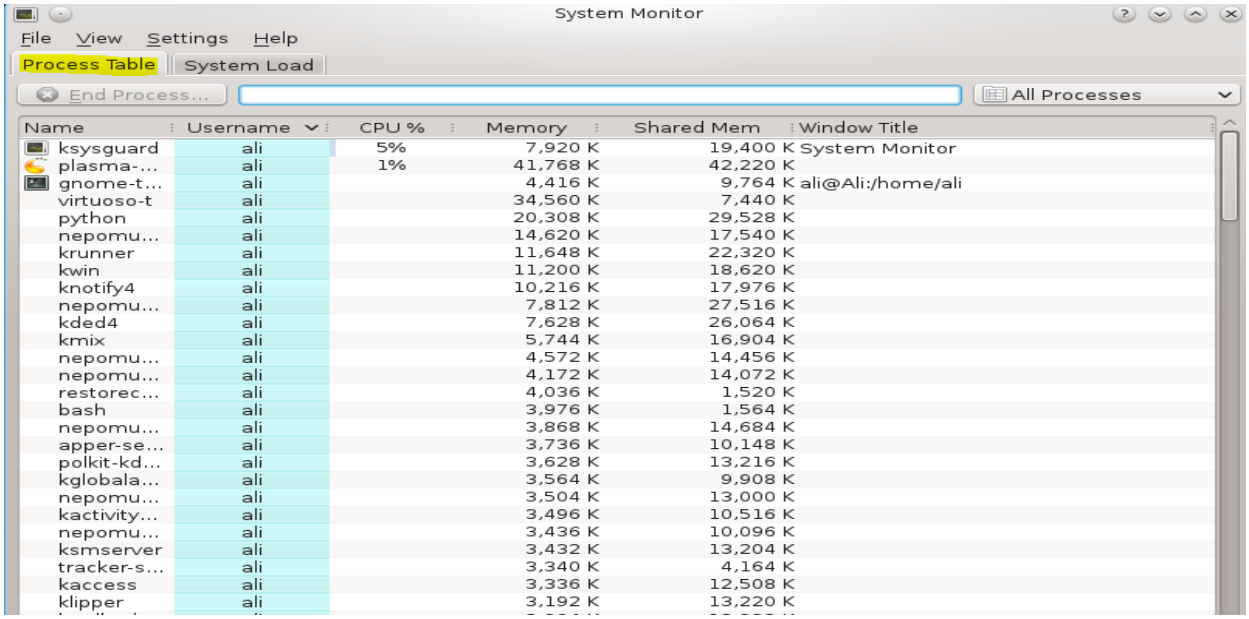
في هذه الطريقة لن تتمكن من إنهاء الإجرائيات التي يملكها مستخدمون آخرون (بما في ذلك إجرائيات النظام) لأنه ليس لديك إذن كافٍ.
من الممكن تشغيل هذا البرنامج كمستخدم root، مما يتيح لك إنهاء أي إجرائية:

```
[ali@Ali ~]$ su root
Password:
[root@Ali ali]# gnome-system-monitor
```

قد يؤدي إنهاء الإجرائية أو الإجرائيات الخاطئة إلى ترك نظام التشغيل في حالة عمل جزئي أو غير قابل للاستخدام، وقد يكون من الضروري إعادة تشغيل النظام للاستعادة لذلك يجب الحذر!

مراقبة الإجرائيات بشكل رسوميّ في بيئة سطح المكتب KDE

من القائمة System نختار KSysGuard (أو بالبحث عن الاسم) وهذا سيشغّل برنامج **ksysguard** الذي سيعرض النافذة التالية:



ملاحظة:

كما هو الحال في GNOME System Monitor، لا يمكن إنهاء الإجرائيات التي يملكها مستخدمون آخرون.
من الممكن تشغيل برنامج **ksysguard** كمستخدم root، مما يتيح لك إنهاء أي إجرائية:

```
[ali@Ali Documents]$ cd
[ali@Ali ~]$ su root
Password:
[root@Ali ali]# sudo ksysguard
```

عرض معلومات عن الإجرائيات بشكل نصيّ (Monitoring process information on a character display)

لدينا الأداة top لعرض تفاصيل كتابية ضمن جدول عن الإجرائيات قيد التشغيل في النظام:

top

```
top - 16:01:18 up 10 min,  2 users,  load average: 0.05, 0.25, 0.21
Tasks: 137 total,   1 running, 136 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  0.3%sy,  0.0%ni, 96.3%id,  2.0%wa,  0.3%hi,  0.7%si,  0.0%st
Mem:   2028592k total,  1462660k used,   565932k free,    43468k buffers
Swap:  4063228k total,        0k used,  4063228k free,   871908k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1782 root      30  10  617m 220m 8880 S  0.7 11.1   0:29.47 yumBackend.py
  910 root      20   0  147m  30m 9540 S  0.3  1.6   0:02.03 Xorg
 1486 root      20   0  171m  18m 4032 S  0.3  0.9   0:01.77 packagekitd
    1 root      20   0 58144  24m 2124 S  0.0  1.3   0:01.26 systemd
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.04 ksoftirqd/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
```

حيث نلاحظ بعض التفاصيل حول الإجرائية مثل المعرّف PID والمستخدم الذي يملكها وأولويتها ومقدار استخدامها للذاكرة والمعالج.

ونبيّن بالجدول التالي بعض الخيارات المتاحة مع الأمر top:

Option	Description
u	**Restrict the display to processes owned by one user.**
M	**Sort by memory usage.**
P	**Sort by current CPU usage.**

يمكنك أيضاً استخدام الأمر ps (process status) لعرض معلومات عن الإجرائية في لحظة زمنية معينة.

بشكل افتراضي، يعرض الأمر ps الإجرائيات التي تم تنفيذها على ال terminalالحالي فقط:

```
[ali@Ali ~]$ ps
  PID TTY          TIME CMD
 1679 pts/0    00:00:00 bash
 2101 pts/0    00:00:00 ps
```

وفيما يلي بعض الخيارات لاستخدامها مع الأمر ps:

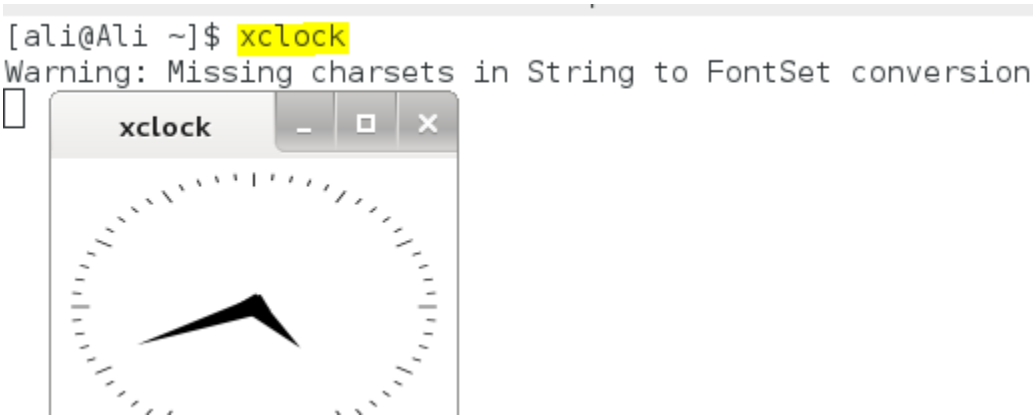
Option	Description
-A -e	All (or everyone's) processes
-u *user*	Processes owned by *user* (which can be a username or numeric user ID)
-f	Displays full information, including the UID, PID, PPID, start time (STIME), terminal (TTY), total CPU time used (TIME), and command (CMD).

إنهاء الإجرائيات باستخدام ال Terminal

يمكنك إنهاء الإجرائيات بواسطة اسم الأمر الذي ولّد الإجرائية أو بواسطة معرّف الإجرائية PID. عندما إنهاء إجرائية باستخدام اسم الأمر، فإنك توفر على نفسك عناء البحث عن المعرّف PID، ولكن إذا كان هناك أكثر من إجرائية واحدة بنفس الاسم قيد التشغيل، فسوف تنهيهم جميعًا.

مثال:

لنكتب الأمر xclock والذي يعرض الساعة مع العقارب ويبقى الأمر قيد التنفيذ:



نفتح terminal آخر ونكتب كالتالي لعرض الإجرائيات ومن ضمنها xclock:

```
[ali@Ali ~]$ ps -u ali
```

ونكتب بعدها كالتالي لإنهاء الأمر السابق:

```
killall xclock
```

إذا لم تنتهي الإجرائية خلال ثوانٍ قليلة، أضف الخيار KILL-.

لاحظ أن هذا سيقتل فقط الإجرائيات بهذا الاسم التي يمتلكها المستخدم الحالي؛ ليس لديك إذن لقتل إجرائيات المستخدمين الآخرين ما لم تكن مستخدم root.

ونكتب الأمر التالي لإنهاء إجرائية عن طريق معرّفها PID (أيضاً إذا لم تنتهي الإجرائية خلال ثوانٍ قليلة، أضف الخيار KILL-):

```
 1679 pts/0    00:00:00 bash
 2208 pts/1    00:00:00 bash
 6618 pts/0    00:00:00 xclock
 6619 pts/1    00:00:00 ps
[ali@Ali ~]$ kill 6618
```

إرسال إشارات للإجرائيات (Sending signals)
لإنهاء إجرائية قيد التنفيذ، يتم إرسال إشارة رقمية إلى تلك الإجرائية.

لعرض جميع الإشارات المتاحة، نستخدم الخيار l- مع الأمر kill، والذي يعرض أسماء وأرقام الإشارات:

```
[ali@Ali ~]$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

وسنذكر فيما يلي بعض هذه الإشارات:

كل من هذه الإشارات لها معنى محدّد، والذي يمكن تحديده عادة من اسم الإشارة.

1- SIGHUP هي إشارة قطع الاتصال. مثلاً عند إغلاق ال terminal يتم إرسال إشارة SIGHUP إلى الإجرائية التي كانت قيد التنفيذ على هذا ال terminal.

2- SIGINT هي إشارة المقاطعة (يتم إرسالها عند استخدام Ctrl-C لمحاولة مقاطعة برنامج يعمل على terminal).

3- SIGPWR هي إشارة إلى انقطاع أو فشل في الطاقة.

ملاحظة:

- يتم إنشاء معظم هذه الإشارات تلقائيًا بواسطة نواة نظام التشغيل.
- في معظم الحالات، يمكن إعداد إجرائية ما للقيام بشيء ما عند التقاط إشارة معينة. إذا لم يتم إعداد الإجرائية عند التقاط الإشارة، فسيتم اتخاذ الحدث الافتراضيّ لهذه الإشارة. في معظم الحالات، يكون الحدث الافتراضي هو إنهاء الإجرائية، ولكن في بعض الإشارات (مثل SIGPWR) يكون الحدث الافتراضي هو ببساطة تجاهل الإشارة والاستمرار في العمل.

4- تستخدم الإشارة **SIGTERM** لتطلب من برنامج معيّن إنهاء نفسه, وهي الإشارة الافتراضية المرسلة من الأمر kill ، وأيضاً هي الإشارة المرسلة بواسطة GNOME System Monitor عندما نضغط على زر End Program.

لإرسال هذه الإشارة باستخدام ال terminal نكتب كالتالي:

kill –SIGTERM *process_id*

5- الإشارة **SIGKILL** غير قابلة للالتقاط والمعالجة وتقوم بإنهاء البرنامج دائمًا بدون إعطائه الفرصة للإغلاق بأمان. يتم إرسال هذه الإشارة عند استخدام الخيار Kill Program ضمن الSystem Monitor .

لإرسال هذه الإشارة باستخدام ال terminal نكتب كالتالي:

kill –SIGKILL *process_id*

6- تُرسل الإشارة **SIGINT** للإجرائية عند حدوث مقاطعة لها, مثلاً عند الضغط على CTRL+C.

مثال:

ليكن لدينا ال script التالي والذي يقوم بالتنفيذ بدون توقّف:

```
#!/bin/bash

while [ true ]; do

echo "on execute ..."

done
```

نفتح terminal آخر ونرسل الإشارة التي نريدها باستخدام الأمر kill واسم الإشارة (**أو رقم الإشارة**) ومعرّف الإجرائية بالشكل التالي(باستخدام الإشارة SIGKILL):

```
 1624 pts/0    00:00:00 bash
 1693 pts/1    00:00:00 bash
 1838 pts/1    00:00:07 nonstop.sh
 1847 pts/0    00:00:00 ps
[ali@Ali ~]$ kill -SIGKILL 1838
```

أو بالشكل التالي(مثلاً باستخدام رقم الإشارة SIGTERM وهو 15):

```
on execute ...
on execute ...
on execute ...
on execute ...
Terminated
[ali@Ali Desktop]$
```

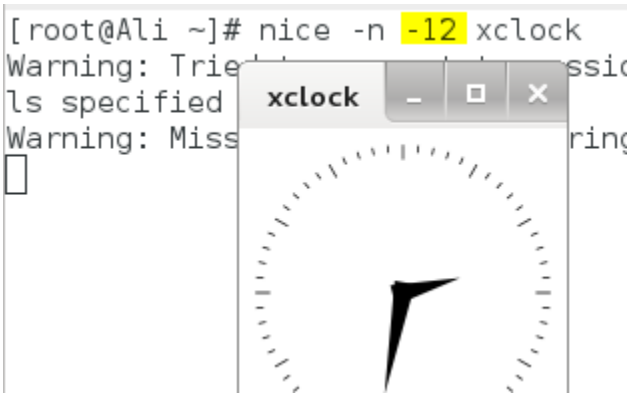
```
 1624 pts/0    00:00:00 bash
 1693 pts/1    00:00:00 bash
 2149 pts/1    00:00:02 nonstop.sh
 2150 pts/0    00:00:00 ps
[ali@Ali ~]$ kill -15 2149
[ali@Ali ~]$
```

تغيير أولوية الإجرائية (nice and renice)

يقوم الأمر **nice** بيدء إجرائية بأولوية أقل من أولويتها المعتادة (lower-than-normal priority).

يمكن تقليل الأولوية بأي قيمة من 1 إلى 19 باستخدام الخيار n-؛ بدون هذا الخيار، يتم تقليل الأولوية بقيمة 10 افتراضياً. بعدها نحدّد الأمر المُراد تشغيله، مثال:

لرفع أولوية الإجرائية، يجب أن تكون مستخدم root؛ وتعديل الأولوية بقيمة سالبة بين -1 (زيادة طفيفة في الأولوية عن الأولوية العادية) إلى -20 (أولوية قصوى):



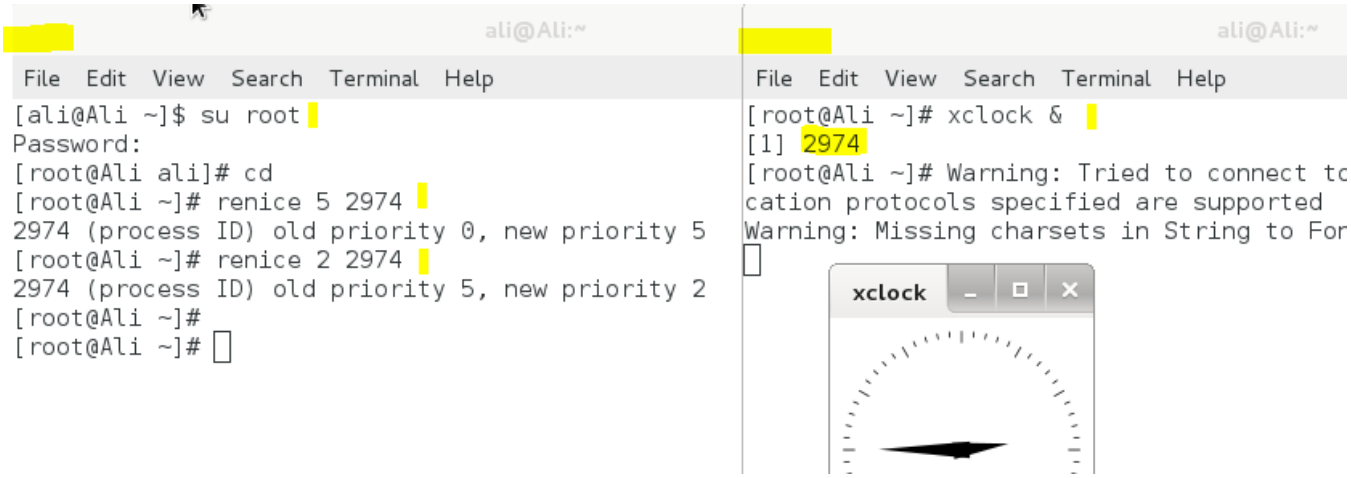
تغيير أولوية إجرائية موجودة وقيد التنفيذ:

نستخدم الأداة (الأمر) **renice** للقيام بذلك.

مثال:

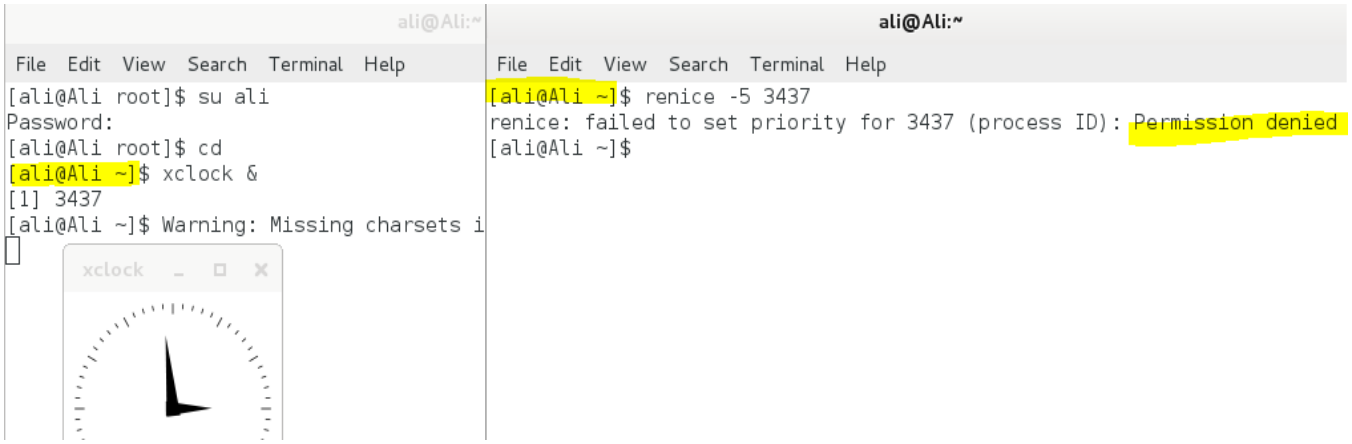
لاحظ استخدامنا للرمز & بعد اسم الأمر من أجل عرض المعرّف PID للإجرائية.

هنا وبعد دخولنا كمستخدم root، واستخدام الأمر renice مع القيمة 5 كوسيط ثانٍ، نكون قد خفّضنا أولوية الإجرائية من 0 إلى 5، بينما بعدها ومع القيمة 2 كوسيط ثانٍ، نكون قد رفعنا أولوية الإجرائية من 5 إلى 2.



ملاحظة:

لا يسمح الأمر renice للمستخدم العادي بزيادة الأولوية لإجرائية معينة حتى لو قام هذا المستخدم بتخفيض الأولوية مسبقاً، بينما يمكن للمستخدم الإداري root أن يعدّل بالأولوية كيفما يشاء:



إدارة المهام المُجدولة (crontab & at)

في Linux، يمكن إعداد المهام لتعمل تلقائيًا خلال فترة زمنية محددة أو في تاريخ محدد، وقد تم إعداد Fedora لتشغيل مهام النظام المهمة للحفاظ على تحديث النظام وعمله.

وقد أتت Fedora مع العديد من أدوات جدولة المهام مثل أداتي cronو at.

- **الأداة cron**

يمكن استخدامها لجدولة تنفيذ المهام الدورية وفق وقت محدد سواء كان شهرياً أو يوم من الشهر أو أسبوعياً أو يوم محدد من الأسبوع.

تفترض الأداة Cron أن النظام يعمل بشكل مستمر. إذا لم يكن النظام قيد التشغيل أثناء جدولة مهمة ما، فلن يتم تنفيذها.

لاستخدام خدمة cron، يجب تثبيت حزمة vixie-cronويجب تشغيل خدمة crond.

لتحديد ما إذا كانت الحزمة مثبتة أم لا، استخدم الأمر rpm -q vixie-cron أما لتحديد ما إذا كانت الخدمة قيد التشغيل، استخدم الأمر

/sbin /service crond status.

- يتكوّن ملف الإعداد الرئيسي للأداة cron والموجود ضمن المسار etc/ / crontab، على الأسطر التالية:

crontab

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

- الأسطر الثلاثة الأولى عبارة عن مُتغيرات تُستخدم لإعداد البيئة التي يتم فيها تشغيل مهام ال cron المُراد جدولتها.
- بعد هذه الأسطر فإنّ كل سطر في الملف etc/ / crontab يُمثّل مهمة معينة ويكون له التنسيق التالي:

```
minute    hour    day    month    dayofweek    command
```

- حيث:

- **minute** – any integer from 0 to 59
- **hour** – any integer from 0 to 23
- **day** – any integer from 1 to 31 (must be a valid day if a month is specified)
- **month** – any integer from 1 to 12 (or the short name of the month such as jan or feb)
- **dayofweek** – any integer from 0 to 7, where 0 or 7 represents Sunday (or the short name of the week such as sun or mon)
- **command** – the command to execute (the command can either be a command such as **ls /proc >> /tmp/proc** or the command to execute a custom script)

- لأي من القيم المذكورة أعلاه، يمكن استخدام علامة النجمة (*) لتحديد جميع القيم المرغوب الجدولة وفقها. على سبيل المثال، إن وضع علامة النجمة لقيمة الشهر تعني تنفيذ الأمر كل شهر.

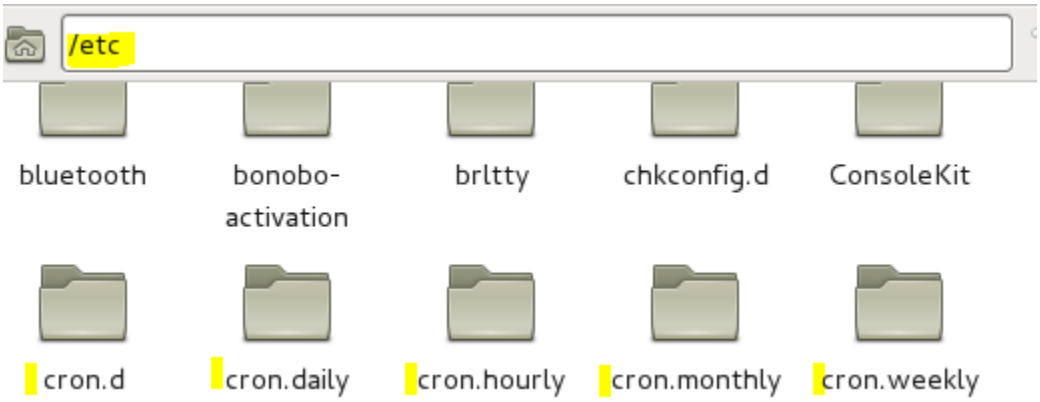
- تحدد الواصلة hyphen (-) بين الأعداد الصحيحة نطاقًا من هذه الأعداد. على سبيل المثال، 4-1 تعني الأعداد الصحيحة 1 و2 و3 و4.

- تحدد مجموعة القيم المفصولة بفواصل (،) قائمة من هذه القيم. على سبيل المثال، 3، 4، 6، 8 يشير إلى تلك الأعداد الصحيحة الأربعة المحددة.

- يمكن استخدام (/) لتحديد قيم الخطوة. يمكن تخطي قيمة عدد صحيح داخل نطاق باتباع النطاق مع / <صحيح>. على سبيل المثال، يمكن استخدام 2 / 59-0 بمعنى كل دقيقتين في حقل الدقيقة. يمكن أيضًا استخدام قيم الخطوة مع علامة النجمة. على سبيل المثال، يمكن استخدام القيمة 3 / * في حقل الشهر لتشغيل المهمة كل شهر ثالث (كلما حل شهر آذار يتم تنفيذ المهمة).

- أي سطور تبدأ بعلامة (#) هي تعليقات ولا تتم معالجتها.

- يتم وضع البرامج النصية(scripts) في المسارات /etc/cron.hourly و /etc/cron.daily و /etc/cron.weekly و /etc/cron.monthly من أجل تنفيذها كل ساعة أو يوميًا أو أسبوعيًا أو شهريًا على التوالي. أي يجب أن تكون الملفات الموجودة في هذه الأدلة عبارة عن نصوص برمجية (scripts).



- إذا كانت مهمة cron مطلوبة ليتم تنفيذها وفقًا لجدول زمني غير كل ساعة أو يوميًا أو أسبوعيًا أو شهريًا، فيمكن إضافتها إلى الدليل /etc/cron.d/.

ملاحظة:

لبدء خدمة cron، استخدم الأمر **/sbin /service cron start** أما لإيقاف الخدمة، استخدم الأمر /sbin / service cron stop.

الأداة at

قلنا إنه يتم استخدام cron لجدولة المهام المتكررة، بينما هنا يتم استخدام الأمر at لجدولة مهمة لمرة واحدة في وقت محدد

لاستخدام at، يجب تثبيت الحزمة في RPM، ويجب تشغيل خدمة atd. لتحديد ما إذا تم تثبيت الحزمة، استخدم الأمر rpm -q at.
لتحديد ما إذا كانت الخدمة قيد التشغيل، استخدم الأمر /sbin / service atd status.

- لجدولة مهمة لمرة واحدة في وقت محدد، اكتب الأمر at time، حيث time هو وقت تنفيذ الأمر.
- يمكن أن يكون الوسيط time واحدًا مما يلي:

- HH:MM format — For example, 04:00 specifies 4:00 a.m. If the time is already past, it is executed at the specified time the next day.
- midnight — Specifies 12:00 a.m.
- noon — Specifies 12:00 p.m.
- teatime — Specifies 4:00 p.m.
- month-name day year format — For example, January 15 2002 specifies the 15th day of January in the year 2002. The year is optional.
- MMDDYY, MM/DD/YY, or MM.DD.YY formats — For example, 011502 for the 15th day of January in the year 2002.
- now + time — time is in minutes, hours, days, or weeks. For example, now + 5 days specifies that the command should be executed at the same time five days from now.

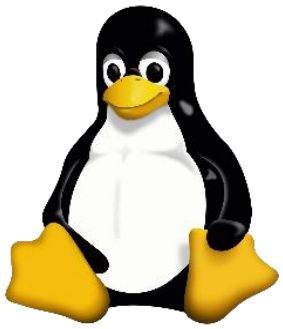
- بعد كتابة الأمر at مع الوسيط time، يتم عرض موجّه أوامر من الشكل at>. بعدها نكتب الأمر المراد تنفيذه، ونضغط على Enter، وبعدها Ctrl + D للخروج.
- يمكن تحديد أوامر متعددة عن طريق كتابة كل أمر متبوعًا بالمفتاح Enter.
بعد كتابة جميع الأوامر، اضغط على Enter للانتقال إلى سطر فارغ واكتب Ctrl + D.
بدلاً من ذلك ، يمكن إدخال نص برمجي كامل (script) في الموجّه ، والضغط على Enter بعد كل سطر في هذا البرنامج النصي ، وكتابة Ctrl + D في سطر فارغ للخروج.

مثال:

كتبنا هنا الأمر at مع تحديد الوقت حتى يتم التنفيذ بعد 3 دقائق من الآن، وبعدها نكتب الأمر المُراد جدولته عند موجّه الأوامر at> ونضغط Enter، وفي حال أردنا أوامر أخرى أو تنفيذ ملفات bash script نكتبها ونتابع ضغط Enter وهكذا إلى أن ننتهي نضغط CTRL+D للخروج:

(لا ننسى أن نشغّل خدمة atd أولاً)

```
[root@Ali ~]# /sbin/service atd start
Redirecting to /bin/systemctl  start atd.service
[root@Ali ~]# at now + 3 minutes
at> ls -l /home/ali
at> <EOT>
job 2 at Sat Nov 28 16:40:00 2020
```



Questions???