

## Run levels

(مستويات التشغيل في لينكس)

ال **runlevel** هو الحالة التي سيعمل عليها النظام.

فمثلا عندما يعمل مدير النظام منفردا دون وجود مستخدمين فهذا **runlevel** ،وعندما يعمل متصلا بالشبكة فهذا **runlevel** آخر، وهكذا.

لتبسيط ال **runlevel** حيث يمكن تشبيهه بال **mode** الموجود في ويندوز مثل **save mode** وغيره .

ول **run level** عدة حالات لكل منها معنى معين ، ولكل حالة رقم يميزها عن بقية ال **run level** الأخرى ، وهي كالتالي :

1. ال **Run level =0 (Halt)** : وهذا يعني أن النظام سيحدث له **halt** (بمعنى **shutdown**)

إذا تم تحويله الى هذا النمط .

2. ال **Runlevel=1 (Single User Mode)** : وهذا يعني أن النظام سيعمل في وضع المستخدم

الواحد ، و المستخدم الواحد هو بالطبع ال **root** ، هذا ال **Runlevel** مخصص لعمليات الصيانة وتنزيل ال **Software** للنظام ، حيث أن هذا النمط لا يحتوي واجهات **GUI**، أي التعامل مع النظام بالكامل يتم عن طريق الأوامر.

3. ال **Runlevel=2 (multiuser without NFS)** : هذا النمط أيضا لا يحتوي واجهات **GUI**

أي يتم التعامل معه باستخدام الأوامر، الا أن في هذا النمط يمكن وجود مستخدمين آخرين مع ال **Root** على النظام و ولكن لا يوجد اتصال للمستخدمين على الشبكة وهذا ما يعنيه ( **Without NFS**) .

4. ال **Runlevel=3 (multiuser with NFS)** : وتعني وجود مستخدمين آخرين مع ال **Root**،

ويمكنهم الاتصال عبر الشبكة ، وهذا النمط لا يحتوي واجهات ( **GUI**) ، السيرفر يعمل في هذا النمط .

5. ال Runlevel=4 (unused) : هذا النمط غير محدد (غير مستخدم) .
6. ال Runlevel=5 (multiuser with graphical) : هذا النمط مشابه للنمط (3) ، الا أنه هذا النمط يحتوي على واجهات (GUI) ، أي يمكن التعامل معه عن طريق الواجهات .
7. ال Run level=6 (Reboot) : عند التحول هذا ال Run level ، فإن النظام سيحدث له إعادة إقلاع .

- يستخدم الأمر ( runlevel ) لمعرفة في أي Level يعمل النظام الآن، كما يلي :

```
ahmad@ahmad-virtual-machine:~$ runlevel
N 5
ahmad@ahmad-virtual-machine:~$
```

- فترى أنه قد ظهر الحرف N وبعده الرقم 5، حيث أن :  
الحرف N يعني أن النظام لم يكن في أي runlevel من قبل .  
أما الرقم 5 يعني أن النظام حالياً في ال runlevel=5 .
- يتم التحول من Runlevel الى آخر باستخدام الأمر init ، وصيغة هذا الأمر :  
**Init <number>** : حيث ال Number هي رقم ال runleve المراد الانتقال اليه .

ملاحظة : ال Runlevels مستويات التشغيل يتم إدارتها عبر ال Init process .

- الملف ( /etc/init/rc-sysinit.conf ) : هذا الملف نحدد فيه النمط الافتراضي ( Default mode ) الذي نريده الإقلاع اليه، عند إقلاع النظام .

نقوم بفتح الملف السابق ، ونقوم باسناد رقم النمط الافتراضي الذي نريده الإقلاع اليه الى الخاصية ( env DEFAULT\_RUNLEVEL ) ، كما يلي :

```
GNU nano 2.2.6      File: /etc/init/rc-sysinit.conf
# rc-sysinit - System V initialisation compatibility
#
# This task runs the old System V-style system initialisation scripts,
# and enters the default runlevel when finished.

description        "System V initialisation compatibility"
author              "Scott James Remnant <scott@netsplit.com>"

start on (filesystem and static-network-up) or failsafe-boot
stop on runlevel

# Default runlevel, this may be overridden on the kernel command-line
# or by faking an old /etc/inittab entry
env DEFAULT_RUNLEVEL=5
```

- المجلدات (/etc/rc(n).d/) : حيث أن n من ال 0 الى ال 6 ، كل واحد من هذه المجلدات مسؤول عن Runlevel معين و يحتوي على مجموعة من ال Scripts والتي تحدد الخدمات التي يمكن أن يقدمها كل Runlevel ، بلغة أخرى يقوم بتنفيذ ال Scripts البرامج الموجودة في ال المجلد (/etc/init.d/) والخاصة بمستوى التشغيل runlevel (n) فقط ، كمثال على ذلك المجلد (/etc/rc2.d) يحدد أي البرامج (scripts) الموجودة في المجلد (/etc/init.d) يجب تفعيلها لتشغيل النمط Runlevel=2، عند بدء التشغيل وإيقاف التشغيل، يتم استدعاء هذه ال Scripts فقط :

```
ahmad@ahmad-virtual-machine:~$ nano /etc/rc
rc0.d/   rc2.d/   rc4.d/   rc6.d/   rcS.d/
rc1.d/   rc3.d/   rc5.d/   rc.local
```

مجموعة المجلدات المسؤولة عن مستويات التشغيل (Runlevels)

```
ahmad@ahmad-virtual-machine:~$ cd /etc/rc0.d
ahmad@ahmad-virtual-machine:/etc/rc0.d$ ls
K10unattended-upgrades  K20speech-dispatcher  S30urandom             S60umountroot
K20kerneloops           README                S31umountnfs.sh        S90halt
K20rsync                 S20sendigs            S40umountfs
```

مجموعة ال Scripts المسؤولة عن ال Runlevel=0

حيث نلاحظ أن أسماء ال Scripts تبدأ بما يلي [k|s]nn :

إما الحرف s وهذا يعني أن هذا ال Script مفعّل من أجل مستوى التشغيل المحدد .

أو الحرف k وهذا يعني أن هذا ال Script غير مفعّل من أجل مستوى التشغيل المحدد .

يلي أحد الحرفين السابقين رقم مكون من خانتين فقط ، وهو عبارة عن رقم تسلسلي يمكن استخدامه للتحكم في تسلسل خدمات البدء، بحيث لا تبدأ الخدمات التي تعتمد على خدمات أخرى إلا بعد بدء تشغيل تلك الخدمات الأخرى .

- المجلد (/etc/init.d/) : يحتوي هذا المجلد على مجموعة من ال scripts تتحكم في إيقاف وتشغيل بعض الخدمات ال Services، وكذلك التحكم في الخدمات في كل مستوى تشغيل من خلال حذفها منه أو إضافتها له ، وذلك باستخدام مجموعة من الأوامر ، حيث :  
1. يستخدم الأمر :

Service (servicename)(process)

حيث أن ال servicename اسم الخدمة التي نريد التحكم بها

ال process وتأخذ القيم التالية :

Status : لمعرفة حالة الخدمة .

Start : لتشغيل خدمة معينة .

Stop : لإيقاف تشغيل خدمة معينة .

Restart : لإعادة تشغيل خدمة معينة .

Reload : يستخدم لإعادة إقلاع الخدمة

مثال : معرفة حالة خدمة الطباعة (cups) ومن ثم إيقاف الخدمة وعرض حالتها مجددا ثم إعادة تشغيلها ،  
كما هو موضح :

```
root@ahmad-virtual-machine:/etc/init.d# service cups status
cups start/running, process 2570
root@ahmad-virtual-machine:/etc/init.d# service cups stop
cups stop/waiting
root@ahmad-virtual-machine:/etc/init.d# service cups status
cups stop/waiting
root@ahmad-virtual-machine:/etc/init.d# service cups start
cups start/running, process 3235
```

2. الأمر **update-rc.d** ، يستخدم هذا الأمر لتفعيل أو تعطيل خدمة معينة في مستوى تشغيل **Runlevel** معين، صيغة الأمر كما يلي :

```
Update-rc.d serviceName enable/disable runlevel
```

مثال : تفعيل خدمة الويب سيرفر ( Apache ) في ال **Runlevel=3** :

```
Update-rc.d apache enable 3
```