

11 - Process Periority & Archieving Compression

Ps -ef

ملفوظة هناك أمر سيب

لا بد منه وجود -

* وهذا الأمر يعرض معلومات عن PID, PPID, UID, CMD

| UID | PID | PPID | C | STime | Time | CMD |
|-----|-----|------|---|-------|------|-----|
| : | : | : | : | : | : | : |

Periority scheduling :-

* لاحظ أنه كل أسبوع يتم عمل backup أي نسخة

من ال Data المتخزنة طول الأسبوع آخرها عندي .

ومن صفات ال backup الناجح أنه أوقف كل ال access

على ال Data أثناء عمل ال backup .

لذلك يتم عمله يوم الخميس مثلاً الساعة ١٢ ص و ٣ ص و ٦ ص

يوم الجمعة . وعندما يخلص ال backup أراجع ال access

على ال data تالية .



لو فرضنا انه ال backup ياخذ 30 min. فال admin

يريد 3 min لى يتم ترجيع ال Access

عنه ال data لذلك يتم ايقاف ال backup اولوية اعلى.

high Periority - حى لو اوقفه كل امكانيات الجهاز.

backup → high Priority

scan for viruses → Low Priority

«كل اسبوع»

على ال Torrogams, viruses

* ولما ال virus scan ياخذ جزء كبير جداً من امكانيات الجهاز

فلو عنى server على web server و FTP server , DNS

نلوا ال virus scan اشغل هيبط الدنيا لذلك اعطى له

Low Periority يعنى خذ وقتك بس متعطش الدنيا.

«والذى يحدد ال Priority هى ال (nice value)

higher Priority

Lower Priority

↑ (minus)

normal
Priority

(Positive) ↓

(-20

19)



وقية ال Priority تكون من (19 ← -20)
 ↓ lower ↓ higher

0 → normal Priority.

nice -n -20
 ↓
 19

application

ex; Ls
 my saldump

CMD syntax

ex; # nice -n 15 Ls -LR /

↓
 Priority لتعديل
 ل APP له هيداً

Low Priority

وفي الطبيعة أن لا نعطى Priority لأمر Ls ولكن لتوضيح الفكرة.

وفي الحقيقة نعطى Priority ل Script أو Application.

ولهذا الأمر في حالة أمر أو Application هيداً له إذا

إذا كان Application شغال أو لا.

renice -n -20 -P Pid

↑
 Priority لتعديل
 ل APP

↑
 numeric
 يتبع له

↓
 19

↓
 Process

شغال.

ولمعرفة ال priority و nice value



لمعرفة ال nice value وال Priority ~ أمر

PS aux ; less

وفيه نتجد عند قائمة ال STAT لو هناك علامة (<)

في high Priority ولو هناك علامة (N) low Priority

(<) → high Priority.

(N) → low Priority.

ولو أردت الحصول على ال nice value كقيمة فيها معنى أمر (TOP)

top

| | | | |
|----------|----|----|------------|
| | PR | NI | |
| | ⋮ | ⋮ | |
| Priority | | | nice value |

والآن هن Run ← Application ونرتب ال nice value بتاعته في ال top.

nice -n -ls -LR /

وهذا الأمر هو أمر (/ -ls -LR) ولكنه هبدأ ب

nice value = -15 أي أننا نطلب تنفيذ الأمر وفي نفس الوقت باشرط

ال Priority الخاصة به قبل تنفيذه.



top



Zombie Process :- i.e. half dead

هو ال Process التي كانت شغالة عندي، وفجأة حصل فيها مشكلة
وليس لل Parent عليها سلطة أو تحكم ولا يستطيع إغلاقها بأي
signal أو أعمل لها reload. وهذا يرجع لل Kernel فلو Kernel
قوى فلا يمكنه أنه تكونه هناك Zombie Ps وعندها تحدث ليس لها حل
إلا عمل reboot.

Archiving & Compressing:-

* والفكرة بينها هو أنه Archiving هو تجميع مجموعة من

النشريات مع بعضها في مكان واحد بدون ضغط.

* أو Compressing هو تجميع مجموعة من الأشياء مع بعضها مع ضغطها

لتصبح أقل في الحجم وفي المساحة.

1) Compressing :-

وعندها فيها طريقتيه كل واحدة لها اسم معين هما:-

1) gzip

2) bzip2



1)

```
# gzip File1 --> File1.gz
```

ولفك الضغط

```
# gunzip File1.gz --> File1
```

2)

```
# bzip2 File1 --> File1.bz2
```

```
# bunzip2 File1.bz2 --> File1
```

وهذا بالنسبة لـ Files العادية.

* لا حظ أنه الزم `bzip2` أفضل من `gzip` لأنهcompression ratio له أعلى من مثيلتها في `gzip`.

* ملاحظ أيضاً أنه الـ rate تباع الـ compression يعتمد على الـ Data

مثل لو ملف الـ `Text` فعند الضغط مساحة تقل جداً أما ملف الـالفيديو فتقل قليلاً فقط عند الضغط. وهو مثل ملف الـ `avi` بالتحديد.

التطبيق العملي

دائماً قبل `create` لـ `File`.

```
# ls -LR / > test
```

ولو عملت

```
# ls -Lh test
```



```
-r-----.1 root root 12M -----test
```

نجد أنه مساحة هذا الملف هي 12 MB

```
# gzip test
```

```
# ls -lh test.gz
```

```
-r-----.1 root root 1.3M -----test.gz
```

وهنا مساحة الـ File المضغوط test.gz أقل بكثير من الملف الأصلي.

لومايز أرفع الملف الأصلي.

```
# gunzip test.gz
```

```
# ls -lh test
```

```
-r-----.1 root root 12M -----test
```

والآن مستخدم أمر `bzip2`:

```
# bzip2 test
```

```
# ls -lh test.bz2
```

```
-r-----.1 root root 926K -----test.bz2
```



وهذه المساحة أقل من نظيرتها في gzip

ولكن المضغوط ←

```
# bunzip2 test.bz2
```



Ls -lh test

نتيجة انه رجع بنفس الحجم بدون اي Loss قبل الحذف.

والآن لننظر الى Archiving

* وهنا آتى بمجموعة من ال files و ال directories

وهنا Archiving ثم Compressing

والآن آتى ب directory كبير و اضعه في المجلد الى انا فيه.

CP -r /usr/ ^{ضيقها في هذا المجلد الى انا فيه}

ولا حظ انه /usr/ هذه مساحتها كبيرة في الاصل.

du usr/ disk usage.

فيأتي بكل File ومساحته.

du -h usr/

وهنا آتى بالقيمة human readable ولما امة الاجالية في آخر التمود.

du -sh usr/ summarized.

لاحظ انه في هناك فرق بين
ex: 131 M usr/ $\frac{1}{8} \text{usr/} \equiv \text{usr}$
in du -sh

والآن عايز اعمل Archiving لهذه الملفات داخل ال /usr/

ويعملها معاملة single file و اضعها



Archieving :- "code syntax"

(tar) يعني أمر لإنشاء

tar cf usr.tar usr/

create file إنشاء Archieving Polder دمج archive

ولفك Archieving

tar xf usr.tar

extract file

tar cf usr.tar usr/

ls -lh usr.tar

-rw-r--r-- 1 root root 129 M ----- usr.tar

لاحظ أنه الملف الناتج من Archieving هو تقريباً 129 م

الأصلية لك directory الأصلي

دالة rm تزيل الملف الأصلي

rm usr -rf




```
# tar xf usr.tar
```

```
# ls -L usr
```

```
usr/  usr.tar
```

وعنه على هذا الأمر يُبين لي
أنه مني ملف Archive
وملف أصلي.

```
# du -sh usr
```

```
131 M  usr
```

مساحة ال directory usr

لاحظ أنه هناك option يس $v \rightarrow \text{verbose}$

يمكنه أنه أيضا ف على options ال Archiving ليكن ~

ملاحظة على ال Archiving وال extracting.

```
# tar cvf usr.tar usr/
```

```
# tar xvf usr.tar
```

والآن ~ مسح الملف usr.tar وهباً لعل Archiving كان.

```
# rm usr.tar
```

```
# tar cvf usr.tar usr/
```

وهنا هيظهر مع التفعيل
مجودة ال File ال على Archive

```
# ls -lh usr.tar
```

نلاحظ المساحة 129M



rm usr -rf ونسخ الملف الأصلي مكانه أنك .

tar xvf usr.tar.

ويظهر في file التي أنشأتها .

du -sh usr

131M usr

إذن في gzip , bzip2 سأتعامل مع file

و tar تتعامل مع Folders و directories وتحولها

إلى Files ومن ثم يمكن استخدامها مع tar أو أدوات

الضغط (gzip & bzip2) compression

tar cvfz usr.tar.gz usr/

ولما استخدمت gzip مع tar وإضافتها تحتها في

tar xvfz usr.tar.gz

tar cvfj usr.tar.bz2 usr/

ولفك الضغط

tar xvfj usr.tar.bz2

ولما استخدمت bzip2 مع tar وإضافتها تحتها في



ولاحظ أنه ال options (x , c) لازم ييجوا في الأول

دبان ال option ترتيبها لا يفرق مني .

تطبيقه على :-

```
# tar cvfz usr.tar.gz usr/
```

لا يترك الاستغناء عنه /

```
# ls -lh usr.tar.gz
```

فيقول أنه ملاحظه (51MB) من الأصل وهو (129MB)

وهو ملاحظه ال usr.tar لوجه لها . بدون ضغط .

والآن نستخدم bzip2 لأنه ال compression ratio له أعلى .

```
# tar cvfj usr.tar.bz2 usr/
```

```
# ls -lh usr.tar.bz2
```

..... 45M ----- usr.tar.bz2

ولفك ضغطه أسمح الملف الأصلي .

```
# rm usr -rf
```

```
# tar xvfz usr.tar.gz
```

```
# du -sh usr
```

131M usr



```
# rm -rf usr
```

```
# tar xvfj usr.tar.bz2
```

ملاحظة:

gzip و bzip2 من نفسهم مسحوا الملف

الأصلي ويتركوا المضغوط فقط. ولكن tar لا
وأخيراً استخدام الضغط مع ال Archiving لا يغيّر الملف الأصلي.

ملاحظة:

```
# tar cvfz usr.tar.gz usr/
```

↓
≡
usr.tar.gz

أي أنه usr.tar.gz ≡ usr.tar.gz

ملاحظة:

لكن أجبنا ال gzip , bzip2 لا يغيروا الملف الأصلي.

```
# ls -lh test
```

* وهذا الملف تحول مسبقاً مساحته 12M

```
# gzip -c test > test.gz
```

```
# ls -lh test
```

وهنا نجد الملف الأصلي مازال موجوداً



الاستجابة:

- 1- منشغل process في ال background و يقول اعمل shell script تعمل search على هذه ال Process وتعلقها.
- 2- عايزين تأخذ backup من /usr/ او من /var/ اى تأخذ نسخة منها ونضيفها ونضع عنها تاريخ اليوم بالساعة والأيام والدقائق.
- 3- shell script تعمل backup ل system.

