

VIM

وثائق أعجوبة

<http://docs.ojuba.org>

رابط الوثيقة : <http://www.ojuba.org/wiki/docs:vim>

ساهم في تحرير هذه الوثيقة :
أسامة عقّاد - كفاح عيسى - مؤيد السعدي

أول تحرير بواسطة أسامة عقّاد بتاريخ 06:19 2008/09/12
آخر تحرير بواسطة أسامة عقّاد بتاريخ 15:07 2008/09/25
تم تصدير هذه الوثيقة بتاريخ : 17:46 2009/12/30



تنويه : تمثل هذه الوثيقة تصديرا لنص على موقع أعجوبة، ولكن رغم ذلك لا يتحمل الموقع أية مسئولية قانونية عن صحة أو خطأ ما يرد فيها.

يسمح لك نسخ أو توزيع أو تعديل هذا المستند
وفق شروط الرخصة الحرة المحددة
حقوق النسخ محفوظة ©

جدول المحتويات

4	1. VIM
4	1.1 المقدمة
4	1.2 البداية
4	1.3 طور الإدخال
4	1.3.1 الإكمال التلقائي
4	1.4 طور الأوامر
4	1.4.1 التنقل في المستند
4	1.4.2 اختصارات العمليات
4	1.4.3 التراجع و النسخ و اللصق
4	1.4.4 تكرار العملية
4	1.4.5 تخزين العمليات
4	1.5 التحديد المرئي
4	1.6 طور التنفيذ أو الامتداد ex
4	1.6.1 البحث
4	1.6.2 الاستبدال
4	1.6.3 الحفظ
4	1.6.4 إضافات
4	1.6.5 مرشد تعليم vim
4	1.6.6 تطبيق عملي
4	1.6.7 مراجع للاستزادة

VIM .1

هذه المقالة تتحدث عن محرر النصوص المتطور VIM ومحرر النصوص الافتراضي في ريدهات لينوكس وغيرها من التوزيعات. هذه مقالة أعدها أسامة عقاد بناءً على مقالة قديمة لمؤيد السعدي

1.1. المقدمة

تم إنشاؤه سنة 1991 على يد برام مولينر **Bram Moolenaar** لحواسيب أميغا **Amiga**. و أنشأ أساساً لتحرير الكود المصدري للبرامج. و يصدر وفق رخصة GPL فهو برنامج حر مفتوح المصدر.

و هو نسخة مطورة عن Vi بالمزيد من التحسينات يمكن تنفيذه من سطر الأوامر باسم vim ويستقبل الخيار A- ليدعم العربية، إذا كان الدعم قد وضع أصلاً عند التصنيف لكن ستظهر الأحرف الإنجليزية مقلوبة و ما زال . ويعمل بكتابة

```
$ vim -A
```

لكن في هذا الشرح سنعتمد النسخة الأصلية بلا دعم العربية

```
$ vim
```

طريقة استعماله لا تختلف كثيراً عن Vi، إذا كنت لم تألف VIM يمكنك تشغيل البرنامج evim الذي يجعل vim يتصرف مثل البرامج العادية السهلة ولكنك ستخسر الكثير من مميزاته وذلك بإضافة الخيار y إلى vim.

له واجهات رسومية مثل gome-vim و kvim لكنها لا تختلف كثيراً عنه شكل vim.

1.2. البداية

كما في Vi تكون في البداية في الطور العادي normal وهو طور العمليات أو الأوامر وفيه تتحرك بالأسهم وغيرها لتحديد المكان الذي تريد تنفيذ العملية به. ولأننا لم نتعلم أيًا منها لننتقل إلى طور آخر.

1.3. طور الإدخال

اضغط i للانتقال إلى طور الإدخال أو الإضافة insertion mode وفي هذا الطور يكون VIM كأى محرر نصوص عادي تتحرك بالأسهم وتطبع النص بلوحة المفاتيح. وهناك العديد من الطرق للدخول في وضع الإدخال حيث تقوم بعمل إضافة للإدخال هي:

- a يدخل النصوص بعد المؤشر
- A يدخل النص في آخر السطر
- i يدخل النصوص قبل المؤشر
- I يدخل النصوص في أول السطر
- O يضيف سطرًا إضافيًا بعد السطر الحالي
- O يضيف سطرًا إضافيًا قبل السطر الحالي

1.3.1. الإكمال التلقائي

من المميزات المفيدة جداً الإكمال التلقائي للنص مثلاً إذ كنت كتبت كلمة internationalization ثم

في مكان ما لاحقاً أردت أن تكتبها اكتب جزء منها مثلاً `inter` ثم اضغط `CTRL+X` للدخول في طور الإكمال التلقائي ثم `CTRL+P` عدة مرات حتى تحصل على الكلمة المطلوبة (في حال كان هناك أكثر من احتمال) أما إذا كان النص الذي تكتبه قبل المكان الموجود فيه أول ظهور للكلمة استعمل `CTRL+N` التي تبحث في مكان لاحق للموضع الحالي. أما إذا كنت تكتب اسم ملف فيمكنك الاكتفاء بكتابة جزء منه وبعد الضغط على `CTRL+X` ثم اضغط `CTRL+F` فيكمل الجزء الب

اقفي فإذا كان هناك أكثر من احتمال يمكنك التنقل بينها ب `CTRL+P` و `CTRL+N` ، مما سيوفر عليك الكثير من العناء. وهناك `CTRL+X` ثم `CTRL+K` التي تكمل النص بناء على قاموس تعدده أنت بشكل مسبق.

1.4. طور الأوامر

نصل له بضغط `ESC` و للتأكد من أننا فيه اضغط `ESC` مرتين.

لأجل التجربة اكتب النص التالي

```
Hello, world! I'm using VIM.
It's cool.
Look this is the 1st statement. And this is the 2nd one. And
there is more.

This is the second paragraph.
```

1.4.1. التنقل في المستند

اضغط `ESC` للعودة للطور العادي أي طور العمليات الآن يمكنك تجربة العمليات التالية التي تستخدم لتحريك المؤشر: `k` سطر للأعلى `z` سطر للأسفل `l` الحرف اللاحق `h` الحرف السابق التي ستظن في البداية أنك لن تستعملها أبداً ، ولكن هناك عمليات أكثر فائدة مثل `0` الذهاب لبداية السطر `^` الذهاب لأول حرف في السطر `$` الذهاب لنهاية السطر `w` الكلمة التالية `e` نهاية الكلمة التالية `b` الكلمة السابقة (الجملة السابقة) `}` الجملة التالية { الفقرة السابقة } الفقرة التالية ربما كتابتها على الشكل التالي تسهل الحفظ:

```
k (up line) w/b word forward/backward
^ e end of word
(prev letter)h <- -> l(next letter) 0/^/$ 1st/1st- nonblank/end of line
\ '/' ' ' prev/next statement
j (down line) '{' '}' prev/next paragraph
```

انتبه: إنه يعامل علامات الترقيم كل كلمة منفصلة مثلاً:

```
He said "I believe in Mohammad"
```

فسيتوقف عند استخدام `w` للانتقال بين الكلمات عند كل من علامتي الاقتباس و لكن عند وجود علامات ترقيم متتابة سيعتبرها كلمة مثل: `"!?"` سيعتبرها كلمة واحدة.

1.4.2. اختصارات العمليات

يمكن إعطاء العمليات رقم لتعمل به (وتكتب قبل العملية) مثلاً `3w` تعني تحرك 3 كلمات للأمام. وهكذا للعمليات التي تليها. لنأخذ المزيد من العمليات مثلاً `x` تحذف الحرف تحت المؤشر بينما `d` مع عملية تحريك تحذف ما بين الموقع الحالي والموقع الذي يصله المؤشر بعد العملية مثلاً `ضع المؤشر على السطر الثالث قبل حرف A من أول And في الجملة الثانية ثم اكتب d` (لتختفي الجملة كاملة ويصبح السطر

```
And this is the 2nd one. And
```

وهذا مفيد إذا كنت تكتب سطر طويل من أكثر من جملة ورأيت أن آخر جملة غير مناسبة يمكنك

حذفها بالضغط على ESC لإعطاء العمليات ثم d للحذف ثم القوس لتحديد من الموقع الحالي إلى بداية الجملة d) ثم تعود لوضع الإدخال وتكتب البديل. ولأنك غالباً تحتاج لوضع كلمة مكان ما حذفت يوفر VIM العملية c (تعني change) التي تقوم بما تقوم به d إلا أنها تدخل في طور الإضافة بعد القيام بالحذف لهذا لو تراجعنا عن آخر عملية وكتبنا مكانها

```
c(fooobar [ESC]
```

سيصبح السطر

```
fooobar And this is the 2nd one. And
```

1.4.3. التراجع و النسخ و اللصق

للتراجع

• للتراجع عن التعديلات u

• للتراجع عن كل التعديلات في السطر الحالي U حيث يقلب بين الأساس و التعديل.

• لعكس التراجع ctrl + r

بالإضافة لميزة التراجع الا محدود فإن VIM يحتفظ بآخر شيء تم حذفه ويمكنك إعادته (لصقه) بالعملية p (أي put) ويمكنك وضعه في غير المكان الذي تم حذفه كما ويمكنك عمل ذلك أكثر من مرة أي إذا ضغطت ppp فإن الكلمة المحذوفة تعود مكررة 3 مرات. أما لنسخ جزء من النص استعمل y التي تعمل كما d و c ولكن دون حذفه أي أنها تحتفظ بالجزء الموجود بين الموقع الحالي للمؤشر والموقع الذي ترسله إليه العملية التالية لها مثلاً y3w تنسخ الكلمات الثلاث التالية ولوضعها في أي مكان آخر قم بتحريك المؤشر إليه ثم اضغط p.

ملحق

c تعديل y نسخ d مسح

• إذا ضغطنا احدها مرتين سيؤثر على سطر مثلاً: CC سيمكنك من تغيير سطر

• إذا أضفنا ا بعد احدها سيؤثر على حرف مثلاً: dl سيحذف حرفا

• إذا ضغطنا w التأثير سيكون على كلمة مثال: yw سينسخ الكلمة.

• p أو P لللصق

لا حظ انه عند نسخ سطر فسيلصق في سطر جديد اسفل موضع المؤشر الحالي و هكذا. قم بتجربة الأوامر بنفسك فالتجربة هنا خير برهان.

تنبيه:

لاحظ أن الأوامر تبدأ من موضع المؤشر و ليس من اول الكلمة فمثلاً إذا كتبت dw سيحذف الكلمة من مكان المؤشر إلى نهاية الكلمة و ليس من بداية الكلمة.

1.4.4. تكرار العملية

يمكنك تكرار آخر تغيير بعملية النقطة . هذه العملية تكرر آخر تغيير في الموقع الحالي وهي لا تحسب عمليات الحركة التي لا تغير. مثلاً إذا وضعت المؤشر عند أول Hello

```
cwWelcome[ESC]
```

التي تعني عَيّر الكلمة الحالية إلى Welcome ثم حركت المؤشر إلى بداية world وذلك بالضغط على w أو بالأسهم ثم ضغطت على النقطة . فإنها هي الأخرى ستصبح Welcome.

1.4.5. تخزين العمليات

يمكنك تخزين 26 فعل معين (مجموعة من العمليات) لتكون اختصاراً macro وذلك بتسجيلها وإعطائها اسماً من أحد الحروف الإنجليزية الـ 26. للبدء بالتسجيل اضغط [ESC] للدخول في الطور العادي ثم اضغط q ثم حرف ليكون اسم الاختصار ثم ابدأ بكتابة الأوامر ثم q أخرى للتوقف عن التسجيل ولتطبيق ما سجلته (ربما في مكان آخر) اضغط في الطور العادي @ واسم الاختصار مثلاً لتسجيل اختصار باسم a يقوم بتغيير الكلمة الثالثة من السطر الحالي لتصبح Hello اكتب

```
[ESC]qa02wcvHello[ESC]q
```

الآن اذهب إلى أي سطر آخر واكتب

```
@a
```

1.5. التحديد المرئي

وهناك طور ثالث في VIM هو التحديد المرئي visual mode وفي هذا الطور تقوم بكتابة عمليات التحريك فيعمل VIM على تحديد منطقة وتظليلها ثم تضغط على رمز العملية للعمل على الجزء المظلل بدلاً من إدخال العملية ثم رمز الحركة كما في الطور العادي. للدخول في هذا الطور يوجد ثلاث طرق أولى هذه الطرق هي الضغط على v انظر إلى السطر التالي

```
This is a very long long line with meaningless english text.
|- ----->
```

لفرض أنك تريد حذف المنطقة المسطرة حرك المؤشر إلى بدايتها ثم اضغط v ثم حرك المؤشر إلى نهايتها واضغط d. لاحظ أنه يوجد طرفين للمنطقة المحددة وأنت تحرك أحدها وللتنقل لتحريك الطرف الآخر اضغط o.

وثاني طرق الدخول إلى الطور المرئي هي بالضغط على V والفرق أن التحديد هنا للسطر كاملاً حتى لو مرت على جزء منه لهذا تسمى linewise visual mode.

وثالثها CTRL+V وهي تسمى blockwise visual mode وهذه تسمح لك بتحديد قطعة مستطيل يفيد عند العمل على عدة أعمدة من الكلمات والرغبة في تعديل عمود منها مثلاً لحذف العمود الخاص بأحمد في

```
ali ahmad belal
exam1 10 8 6
exam2 8 8 7
exam3 9.5 7 6
```

ضع المؤشر على حرف a من ahmad ثم ادخل هذا الطور بالضغط على CTRL+V ثم تحرك إلى نهاية الكلمة ب e ثم انزل 4 أسطر ب z أو بالأسهم ثم اضغط d لحذف ما تم تعليمه. ونلاحظ أن المنطقة المحددة هنا لها 4 أطراف تنتقل بينها بالضغط على o أو O. اكتب نصاً كالتالي

```
#include <gl.h>
#include <glu.h>
#include <glut.h>
#include <glui.h>
```

لفرض أنك تريد تحويل أسماء الملفات إلى gl/gl.h و gl/glu.h و gl/glut.h و gl/glui.h ضع المؤشر عند حرف g الأول من gl.h ثم اضغط CTRL+V ثم انزل لتحديد كامل الأسطر ثم اضغط

```
lg/[ESC]
```

حيث a هي عملية إضافة النص g/ قبل المنطقة المحددة. لاحظ أنك يجب أن لا تنس [ESC] وأنت قبل الضغط عليها لن ترى الإضافة إلا في سطر واحد.

1.6. طور التنفيذ أو الامتداد ex

في هذا الطور تتم عمليات البحث و الاستبدال و الحفظ. للدخول لهذا الطور نكتب علامة النقطتين من طور الأوامر :

1.6.1. البحث

يمكن الانتقال إلى أول حدوث لحرف معين باستعمال f أو لبحث للخلف ب F متبوعتان بالحرف المطلوب ويمكن تكرار هذا البحث بالفاصلة المنقوطة ; للبحث المتطور يوفر VIM ميزة التعرف على التعبيرات العادية Regular Expressions (للاختصار RE) وهي طريقة قوية للبحث تحتوي على رموز تمثل معنى غير معناها الحقيقي مثلاً | التي تعني أو انظر فصل "العودة لسطر الأوامر". فإذا كنت تبحث عن بريد email لشخص يكون هو نفسه الموقع مثل ahmad@ahmad.com فإن ال RE هي

```
\\([A-Za-z0-9]\\+\\)@\\1\\.com
```

للبحث عن تعبير معين للأمام / ثم أدخل التعبير أما للخلف ? و لتكرار آخر بحث n, وبطريقة عكسية .shift + n

1.6.2. الاستبدال

للبحث عن تعبير عادي وإبداله بشيء آخر نستعمل substitute: أو s: ويمكنك تحديد الأسطر التي تريد العمل عليها مثلاً من سطر 2 إلى 15 بكتابة

```
:2,15s
```

فإذا أردت النص كله استعمال % أي

```
:%s
```

متبوعة بالتعبير الذي تريد البحث عنه ثم العبارة الجديدة التي يضعه مكانه ثم خيارات يفصل بينهما / مثلاً

```
:%s/Ali/Ahmad/gc
```

أما الخيارات المستعملة فيمكن أن تكون g أي أكثر من مرة في السطر ويمكن أن تكون c أي أن يسأل قبل كل تغيير. لاحظ أنه يمكنك الاستفادة من كامل مزايا RE مثل أن تكون العبارة كلمة مستقلة وليس جزء من كلمة باستعمال <> مثل

```
:%s/\\<Ali\\>/Ahmad/gc
```

أو (وهذا المثال من وثائق VIM) إذا كنت تريد إبدال كل عبارتين تفصل بينهما فاصلة مكان بعض بأن تصبح bar , foo بعد التعديل bar foo بالأمر

```
:%s/\\([^\,]*\\), \\(.*\\)/\\2 \\1/
```

1.6.3. الحفظ

بكتابة علامة النقطتين : ثم نكتب الاوامر

w• حفظ

wq• حفظ و خروج

q• خروج (في حال عدم التعديل على الملف)


!q• خروج مع تجاهل التعديلات

e• إعادة النص لما قبل التعديل

! تضاف هذه العلامة بعد الأوامر للإجبار عليها. مثل حالة الخروج دون حفظ التعديلات لن يرضى بـ q لوحدها. وعند ضغط زر الإدخال سيتم تنفيذ الأمر.


1.6.4. إضافات

يمكنك عند تشغيل vim إضافة الخيارات التالية

- m- يمنع تعديل الملف
 - R w- يمنع التعديل إلا بتأكيد التعديل و الحفظ ب! 
 - x- يشفر الملف
 - n- لا يستعمل ملف swap
 - r- استعادة ملف من swap
- مثال:

```
vim -x hello
```

1.6.5. مرشد تعليم vim

vimtutor برنامج يأتي مع vim حيث يعطي المستخدم فكرة عن استخدام vim و يقدم حالات عملية و أمثله بشكل دروس مقسمة إلى فصول مرتبه. حيث يهدف هذا المرشد إلى إعطائك معلومات كافية لجعل vim محررك لكافة الأغراض. و يحوي هذا البرنامج 7 دروس و تتراوح بين النسخ واللصق و طريقة تنفيذ أوامر الصدفة  من داخل البرنامج و حتى طريقة حفظ الملفات أو قسم من النص. يتم تشغيله بكتابة

```
vimtutor
```

في سطر الأوامر.

1.6.6. تطبيق عملي

1.6.7. مراجع للاستزادة

الموقع الرسمي لـ VIM

www.vim.org

من الموقع الرسمي ستجد روابط لكل ما تريد عن VIM نصائح ,نصوص برمجية ,أخبار ,وثائق...إلخ.

• ويكي لتبادل النصائح و الخدع

• وثائق VIM

• صفحة VIM في ويكيديا

• كتاب PDF لـ ستيف اولين عن VIM

• نصوص برمجية scripts

• الأسئلة الشائعة

• المزيد من الروابط

للمساعدة من داخل البرنامج:

```
:help
```