جامعة حمص
الكلية التطبيقية
قسم تقنيات الحاسوب نظم التشغيل الشبكية -1-
المحاضرة الثالثة

محرر النصوص VI

ما هو محرر النصوص VI :

هو المحرر الافتراضي لنظم تشغيل Unix ويدعى VI اختصاراً لكلمتي (Visual Editor) كما وأنه المحرر البديل لنظم التشغيل التي تعمل بشكل مشابه لأنظمة Unix مثل Emacs التي قامت GNU بإنتاجه.

هو محرر يعمل على كامل الشاشة وفق أحد النمطين :

1) النمط الأول : نمط الأوامر Command Mode والذي يقوم بتطبيق أوامر على الملفات.
2) النمط الثاني : نمط الإدخال Insert Mode وفيه يتم وضع النص المدخل ضمن الملف.

في نمط الأوامر فإن كل محرف يتم إدخاله يعبر عن أمر يطبق على الملف النصي المحرر ولإنهاء هذا النمط يتم إدخال محرف محدد فيقوم بتغيير نمط VI من نمط الأوامر إلى نمط الإدخال وهنا تبدأ عملية كتابة النص المحتوى للملف النصي المراد إنشاؤه أو تعديله.

في نمط الإدخال فإن كل محرف مدخل تتم إضافته إلى محتوى الملف النصي ولإنهاء هذا النمط نقوم بالضغط على Esc.

أوامر VI :

للتعامل مع محرر النصوص VI نحن بحاجة لمجموعة من الأوامر للتعامل معه وهنالك الكثير منها وسنتعرف على مانحتاجه للبدء بالعمل على محرر النصوص.

1) **لبدء VI :** نقوم بفتح تطبيق Terminal ثم نكتب الأمر :

vi filename

حيث filename هو اسم الملف المراد فتحه (مسار الملف) وهنا إما أن يكون الملف موجوداً فيتم فتحه وعرض محتواه أو أن يكون غير موجود فيتم إنشاؤه وفتحه ليتم التعديل عليه.

vi filename	لبدء محرر النصوص VI وتحرير الملف المحدد اسمه
vi –r filename	لاستعادة الملف المحدد اسمه والمحتوى في حال أغلق النظام بطريقة غير اعتيادية (Recovery)
View filename	لفتح ملف موجود مسبقاً في وضع القراءة فقط

2) لإغلاق VI :

عادة بمجرد فتح VI بالنسبة لملف معين فيتم حفظ الملف والتعديلات، ولكن إذا أردنا إغلاق VI دون حفظ الملف أو مع بعض الخصائص الأخرى يمكن اختيار أحد هذه الأوامر مسبوقة بالرمز (:) ونضغط على زر Enter.

:x <Enter>	الخروج من VI وحفظ التعديلات على الملف المحدد بمكانه الأصلي
:w <Enter>	لحفظ التعديلات على الملف المحدد بمكانه الأصلي
:wq <Enter>	الخروج من VI وحفظ التعديلات على الملف المحدد بمكانه الأصلي
:q <Enter>	الخروج من VI
:q! <Enter>	الخروج من VI حتى لو لم يتم حفظ التعديلات الأخيرة على الملف

3) تحريك مؤشر الكتابة :

عادة لتحريك مؤشر الكتابة لابد من استخدام أوامر محددة حيث لايمكن التحكم بالمؤشر عن طريق أزرار الأسهم لأن نسخة المحرر مبنية على أساس لوحة مفاتيح لا تحوي الأسهم وعادة مايعطي استخدام الأسهم أوامر غريبة أو تنفيذات خاطئة وبالرغم من ذلك قد نصادف نسخ من نظم التشغيل التي تسمح باستخدام الأسهم وتقوم بتحريك المؤشر عن طريقها.

الأوامر المسؤولة عن تحريك المؤشر :

لتحريك المؤشر للأسفل	J (or Enter)
لتحريك المؤشر للأعلى	K
لتحريك المؤشر نحو اليسار	H
لتحريك المؤشر نحو اليمين	L
تحريك المؤشر للسطر الأول في الملف	:0 <Enter>
تحريك المؤشر للسطر الذي رقمه n (حيث n هنا هي رقم السطر)	:n <Enter>
تحريك المؤشر للسطر الأخير من الملف	:$ <Enter>

الأوامر التالية تقوم بتحريك المؤشر عدة أسطر للأعلى أو للأسفل:

يقوم بتقليب الشاشة بشكل كامل للتي تليها	^F
يقوم بتقليب الشاشة بشكل كامل للتي تسبقها	^B
إعادة عرض الشاشة	^L

حيث الرمز ^ يشير إلى ضرورة ضغط زر ctrl بشكل مستمر عند تنفيذ الأمر

إضافة، تعديل، وحذف نص :

على خلاف كل محررات النصوص التي استخدمناها لايمكن التعامل مع النصوص بمجرد تحديدها والتعديل عليها وإنما نحن بحاجة لمجموعة من الأوامر كما هو الحال في فتح المحرر أو إغلاقه أو تحريك المؤشر ضمن الملف النصي المفتوح.

والأهم من هذا كله أنه لا يمكنك العودة بالتنفيذ والتراجع إلا خطوة واحدة فقط للخلف وفق الأمر التالي:

التراجع عن آخر عملية تم تنفيذها	U

4) إدخال أو إضافة نص :

سنعرض فيما يلي مجموعة من الأوامر التي تسمح بإدخال وإضافة نص إلى الملف وكل من هذه الأوامر تنقل المحرر من نمط الأوامر إلى نمط الإدخال، وللخروج منه والعودة لنمط الأوامر لابد من الضغط على زر Esc للعودة إلى نمط الأوامر.

إدخال النص قبل المؤشر (حتى نضغط على زر Esc)	i
إدخال النص من بداية السطر الحالي (حتى نضغط على زر Esc)	I
إضافة نص بعد المؤشر (حتى نضغط على زر Esc)	a
إضافة نص لنهاية السطر الحالي (حتى نضغط على زر Esc)	A
بدء وإدخال النص بسطر جديد قبل السطر الحالي (حتى نضغط على زر Esc)	o
بدء وإدخال النص بسطر جديد بعد السطر الحالي (حتى نضغط على زر Esc)	O

5) تغيير النص :

الأوامر التالية هي المسؤولة عن تغيير النص والتعديل عليه

r	تغيير محرف واحد عند المؤشر (لا حاجة لضغط Esc)
R	تغيير مجموعة محارف بدايتها المحرف الحالي عند المؤشر ويستمر التعديل حتى نضغط زر Esc
cw	تغيير كلمة كاملة بدءاً من المؤشر ويستمر التعديل حتى نضغط زر Esc
cNw	تغيير N كلمة بدءاً من أول محرف عند المؤشر ويستمر التعديل حتى الضغط على زر Esc
C	تبديل أو تغيير المحارف في السطر الحالي بأكمله حتى الضغط على زر Esc
.cc	تغيير كل السطر الحالي بمدخلات جديدة حتى الضغط على زر Esc
Ncc or cNc	تغيير N سطراً من السطور التالية بدءاً من المحرف الحالي عند المؤشر حتى الضغط على زر Esc

6) حذف النص :

الأوامر التالية مسؤولة عن حذف محرف أو مجموعة محارف أو نص بأكمله

X	حذف المحرف الحالي عند المؤشر
Nx	حذف مجموعة محارف عددها N بدءاً من المحرف الحالي عند المؤشر
dw	حذف كلمة واحدة تبدأ عند المحرف الحالي للمؤشر
dNw	حذف N كلمة بدءاً من المحرف الحالي عند المؤشر
D	حذف تتمة السطر الحالي بدءاً من مكان المؤشر وحتى نهاية السطر
dd	حذف السطر بأكمله من بدايته وحتى نهايته
Ndd or dNd	حذف مجموعة أسطر عددها N بدءاً من السطر الحالي الذي يقع فيه المؤشر

7) نسخ ولصق نص :

أهم الأوامر المسؤولة عن عملية النسخ واللصق

yy	نسخ السطر الحالي إلى الذاكرة
Nyy or yNy	نسخ N سطراً التاليين بدءاً من موقع المؤشر ووضعه في الذاكرة
p	لصق ماهو منسوخ في الذاكرة إلى السطر التالي بعد السطر الحالي

8) أوامر البحث في النص :

ماهو شائع في عملية البحث هو عملية الاستبدال أكثر من عملية البحث بحد ذاتها أي نقوم باستبدال مجموعة محارف أو سلسلة نصية بأخرى، وبالتالي لدينا مجموعة الأوامر التالية

/string	البحث عن تطابق للسلسلة string في النص التالي
?string	البحث عن تطابق للسلسلة string في النص السابق
n	الانتقال إلى التطابق التالي للسلسلة (بعد أن نطبق أمر البحث)
N	الانتقال إلى التطابق السابق للسلسلة (رجوع) (بعد تنفيذ أمر البحث)

9) **تحديد عدد الأسطر :**

إن عملية تحديد عدد الأسطر المتبقية في النص بعد السطر الحالي أو العدد الكلي للأسطر التي تم تحريرها يعتبر من الأمور الهامة والمفيدة والتي قد نحتاجها ولذلك لدينا مجموعة الأوامر التالية

يعيد عدد الأسطر المتبقي من السطر الحالي وحتى نهاية الشاشة فقط	=. :
يعيد العدد الكلي للأسطر حتى نهاية الشاشة الحالية	:=
يعيد رقم السطر الحالي مع عدد الأسطر الكلي للملف	^g

10) **قراءة وحفظ الملف :**

بالنهاية نحن بحاجة لقراءة الملف النصي وفتح ملف آخر للقراءة، وقد نحتاج لدمج محتوى الملف الجديد مع الملف المفتوح ومن ثم حفظه. الأوامر التالية تسمح لنا بالقيام بهذه العمليات

فتح الملف المحدد اسمه filename وإضافة محتواه للملف الحالي بدءاً من السطر الحالي (السطر الذي فيه المؤشر)	:r filename <Enter>
كتابة المحتوى الحالي ضمن ملف جديد اسمه newfile	:w newfile <Enter>
كتابة المحتوى الموجود بدءاً من السطر 12 وحتى السطر 35 في ملف جديد اسمه smallfile	:12,35w smallfile <Enter>
كتابة المحتوى الحالي على ملف موجود مسبقاً مع حذف محتواه السابق حيث اسم الملف هو prevfile	:w prevfile <Enter>

إعادة التوجيه

ما المقصود بإعادة التوجيه ؟

هي عملية إعادة توجيه الدخل والخرج أي I/O (اختصاراً لـ Input/Output) بحيث يمكننا التحكم بالدخل والخرج من الملفات إلى الشاشة السوداء أو من الشاشة السوداء إلى الملفات، أي التحكم بما سيتم عرضه على الشاشة من الملف أو ما ستتم كتابته من الشاشة للملف.

ما هي الأوامر المستخدمة في إعادة التوجيه ؟

1) cat: جمع الملفات.
2) sort: ترتيب الأسطر النصية.
3) uniq: التبليغ عن (أو حذف) الأسطر المتكررة.
4) grep: عرض الأسطر التي تطابق أمراً محدداً.
5) wc: عرض عدد الأسطر والكلمات وعدد البايتات في ملف.
6) head: عرض القسم الأول من الملف (السطور الأولى).
7) tail: عرض القسم الأخير من الملف (السطور الأخيرة).

كيف تتم عملية إعادة التوجيه ؟

عادة عندما نتعامل مع أي برنامج أو حتى مع أوامر لينوكس فإن ناتج تنفيذ هذه الأوامر ينقسم إلى قسمين : قسم يمثل الناتج (الخرج القياسي) وقسم يمثل رسائل الخطأ في حال حدوث أي خطأ أثناء التنفيذ (خرج الخطأ) ونحن عادة نشير للخرج القياسي بكلمة stdout ولخرج الخطأ بكلمة stderr وكلاهما يرتبط افتراضياً بالشاشة السوداء (الطرفية).

بالمقابل كذلك الأمر بالنسبة للدخل لدينا مجرى دخل قياسي ندعوه stdin يرتبط افتراضياً بلوحة المفاتيح.

تسمح آلية إعادة توجيه الدخل والخرج بتغيير المكان الذي سيذهب إليه الخرج والمكان الذي سيأتي منه الدخل عموماً فبدل أن يذهب الخرج إلى الشاشة والدخل من لوحة المفاتيح سيكون هنالك مجاري مختلفة تماماً لتأمين بيانات الخرج أو الدخل.

إعادة توجيه مجرى الخرج القياسي :

تسمح لنا هذه الآلية بتغيير مجرى الخرج وتحديد المكان الذي ستذهب إليه مخرجات الأوامر (أي ملف آخر عدا الشاشة) ويرمز لإعادة التوجيه بالرمز (>) ومن ثم يتبعه مسار الملف المراد إعادة التوجيه إليه.

مثال: لو أردنا توجيه مخرجات الأمر ls (أي معلومات عن محتويات المسار /usr/bin) من الطرفية ليتم تخزينها في ملف نسميه ls-output.txt بدلاً من ظهور ناتج تنفيذ الأمر على الشاشة

```
[me@linuxbox ~]$ ls -l /usr/bin > ls-output.txt
```

ملاحظة: في حال كان الملف الذي سنقوم بإعادة التوجيه إليه موجوداً مسبقاً فستتم الكتابة فوقه، وفي حال لم يكن موجوداً سيتم إنشاؤه، أما لو أردنا أن تتم الكتابة دون مسح محتوى الملف (أي أن تضاف إلى محتويات الملف الأصلية) عندها نستخدم (>>) بدلاً من (>) لإعادة التوجيه.

مثال: لو أردنا أن ندمج خرج الأمر ls للمسار /usr/bin مع محتوى الملف ls-output.txt بدلاً من مسحه والكتابة فوقه

```
[me@linuxbox ~]$ ls -l /usr/bin >> ls-output.txt
```

إعادة توجيه مجرى الخطأ القياسي :

لو أردنا تنفيذ الأمر السابق ls على مسار غير موجود أصلاً ستعرض رسالة خطأ على الشاشة تخبرنا بعدم وجود المسار المحدد وفق مايلي :

```
[me@linuxbox ~]$ ls -l /bin/usr > ls-output.txt
ls: cannot access /bin/usr: No such file or directory
```

لو أردنا إعادة توجيه هذه الأخطاء (بدلاً من أن تظهر رسالة الخطأ على الشاشة سنقوم بتخزينها في ملف على سبيل المثال) نستخدم الرمز (2>) يليها مسار الملف الذي سيخزن فيه رسائل الخطأ فهي تخبر الطرفية في حال حدوث أي خطأ يجب توجيهه لمسار الملف المحدد بعد الرمز السابق

```
[me@linuxbox ~]$ ls -l /bin/usr 2> ls-error.txt
```

ملاحظة: لو أردنا توجيه مجرى الخرج القياسي ومجرى الخطأ القياسي سوية نستخدم الرمز (&>) وهو مسؤول عن توجيه الخرج أو الخطأ (بذلك نكون قد راعينا حالات الخطأ بحيث لا تظهر على الشاشة) كما في المثال التالي:

```
[me@linuxbox ~]$ ls -l /bin/usr &> ls-output.txt
```

التخلص من المخرجات :

لو أردنا التخلص من المخرجات (أي عدم ظهور أي خطأ أمام المستخدم على الشاشة) بحيث نتخلص من رسائل الخطأ دون تخزينها أو الاحتفاظ بها، نقوم بما يلي :

```
[me@linuxbox ~]$ ls -l /bin/usr 2> /dev/null
```

حيث أن /dev/null هو عقدة فارغة، أي أنها تقبل المخرجات دون أن تقوم بأي عمل بها وإنما تقوم فيما بعد بإتلافها.

إعادة توجيه مجرى الدخل القياسي :

هنا بدلاً من أن تقبل الطرفية البيانات من مجرى الدخل القياسي (المرتبط افتراضياً بلوحة المفاتيح) ستقبل الطرفية هذه البيانات من ملف ما مخزن.

1) الأمر cat :

يستخدم هذا الأمر لإعادة توجيه مجرى الدخل القياسي، فبدلاً من أن تتم طباعة البيانات من لوحة المفاتيح سيتم جلبها من الملف المحدد مساره بعد هذا الأمر.

مثال: لو أردنا عرض بيانات الملف ls-output.txt

```
[me@linuxbox ~]$ cat ls-output.txt
```

ملاحظة: طالما أن الأمر cat يقبل عدة ملفات مع بعضها البعض فهو قادر على ضم هذه الملفات مع بعضها البعض لتصبح ملف واحد، في المثال التالي لدينا الملفين t1.txt و t2.txt ونريد دمجهما في الملف t.txt

```
cat /usr/bin/t1.txt /usr/bin/t2.txt > /usr/bin/t.txt
```

يمكن استخدام الأمر السابق لإنشاء ملف والكتابة فيه أو فتح ملف موجود مسبقاً والكتابة فيه وفقاً لما يلي :

```
[me@linuxbox ~]$ cat > lazy_dog.txt
The quick brown fox jumped over the lazy dog.
```

ونستمر بالإدخال والتخزين في الملف lazy_dog.txt حتى نضغط ctrl+d والتي تخبر الطرفية بإنهاء الملف وحفظ البيانات المدخلة.

2) الأنابيب :

نستخدم الأنابيب (|) بين أمرين متتاليين لتمرير مجرى الخرج القياسي للأمر الأول إلى مجرى الدخل القياسي للأمر الثاني، أي أن ناتج تنفيذ الأمر الأول سيكون دخل للأمر الثاني.

الصيغة العامة : `command1 | command2`

3) الأمر sort :

يستخدم هذا الأمر لترتيب الأسطر النصية، ويعتمد على الترتيب الأبجدي بحيث يقوم بترتيب الأسطر التي تبدأ بأرقام أولاً ثم تليها الأسطر التي تحوي المحارف وهكذا..

الصيغة العامة لترتيب أسطر الملف ترتيباً أبجدياً	sort filename
يقوم بعرض الأسطر مرتبة ترتيباً عكسياً	sort –r filename
يقوم بتغيير مجرى الخرج القياسي ليتم تخزين الناتج في ملف outputfile	sort –o outputfile filename
يقوم بترتيب الأسطر اعتماداً على قيم عمود معين في حال كانت بيانات الأسطر عبارة عن أعمدة مفصولة فيما بينها بفراغ	sort –k filename

مثال: سنقوم بعرض محتويات ملف ما بالترتيب الأبجدي كما يلي :

أولاً سنقوم بإدخال محتويات الملف كما يلي :

```
$ cat > mix.txt
abc
apple
BALL
Abc
bat
```

ثانياً سنقوم بعرض محتويات الملف مرتبة ترتيباً أبجدياً :

```
$ sort mix.txt
Output :
abc
Abc
apple
bat
BALL
```

مثال:

```
$ sort file.txt > output.txt
$ sort -o output.txt file.txt
$ cat output.txt

Output :
abhishek
chitransh
divyam
harsh
naveen
rajan
satish
```

مثال:

```
$ cat > employee.txt
manager  5000
clerk    4000
employee  6000
peon     4500
director 9000
guard     3000
```

الآن سنقوم بترتيب الأسطر اعتماداً على العمود الثاني :

```
$ sort -k employee.txt
guard    3000
clerk    4000
peon     4500
manager  5000
employee 6000
director 9000
```

وبالتالي في حال قمنا بتنفيذ الأمر ls لعرض معلومات عن محتويات مسارين معاً على سبيل المثال فسنحصل على قائمتين كل قائمة مرتبة على حدى، ولو أردنا ترتيب هاتين القائمتين لتصبحا قائمة واحدة مرتبة فسنحتاج لاستخدام الأنابيب بحيث يكون خرج الأمر ls هو دخل للأمر sort كما يلي :

```
[me@linuxbox ~]$ ls /bin /usr/bin | sort
```

هنا سيتم تنفيذ الأمر ls على كل من المسارين bin/ و usr/bin/ فيكون لدينا قائمتين مرتبتين وقمنا بتوجيه الناتج عن طريق الأنبوب واستخدمنا الأمر sort للحصول على قائمة واحدة مرتبة.

4) **الأمر uniq :**

يقوم هذا الأمر بحذف الأسطر المكررة افتراضياً وهو يقبل الدخل من مجرى الدخل القياسي (المرتبط بلوحة المفاتيح افتراضياً) أو من ملف أو من ناتج تنفيذ أمر ما.

uniq –option [input]	الصيغة العامة للأمر وهي تقوم بحذف المتكرر
uniq filename	تقوم بعرض محتويات الملف بعد حذف الأسطر المتكررة
uniq –c filename	تعرض في بداية كل سطر عدد مرات تكراره
uniq –d filename	تعرض الأسطر المتكررة فقط
uniq –u filename	تعرض الأسطر غير المتكررة فقط

مثال: ليكن لدينا الملف myfile.txt والذي يحتوي على :

```
This is a line.
This is a line.
This is a line.

This is also a line.
This is also a line.

This is also also a line.
```

الآن سنقوم بتنفيذ الأمر uniq بخياراته المختلفة :

```
uniq myfile.txt
```

```
This is a line.

This is also a line.

This is also also a line.
```

```
uniq -c myfile.txt
```

```
3 This is a line.
1
2 This is also a line.
1
1 This is also also a line.
```

```
uniq -d myfile.txt
```

```
This is a line.
This is also a line.
```

```
uniq -u myfile.txt
```

```
This is also also a line.
```

الآن يمكن تغيير مجرى الدخل لهذا الأمر فبدل أن يكون ملف يمكن أن نستخدمها لحذف التكرار من ناتج تنفيذ أمر ما وليكن الامر ls على سبيل المثال.

مثال: لو أردنا تنفيذ الأمر ls بالنسبة لمسارين وترتيب القائمتين لنحصل على قائمة واحدة ومن ثم حذف التكرارات في هذه القائمة وعرضها، نكتب:

```
[me@linuxbox ~]$ ls /bin /usr/bin | sort | uniq
```

لو أردنا عرض المتكرر فقط في القائمتين لا بد من ترتيب القائمتين لتصبحا قائمة واحدة ومن ثم عرض المتكرر فقط:

```
[me@linuxbox ~]$ ls /bin /usr/bin | sort | uniq -d
```

5) الأمر wc :

يستخدم الأمر wc (وهو اختصار لـ word count) لحساب عدد الأسطر والكلمات والبايتات في الملفات.

الصيغة العامة	wc –options filename
عدد أسطر الملف	wc –l filename
عدد كلمات الملف	wc –w filename
عدد محارف الملف	wc –m filename
يعرض طول أطول سطر في الملف	wc –L filename

مثال: لو أردنا عرض عدد الأسطر في الملف tecmint.txt نكتب:

```
[root@tecmint ~]# wc -l tecmint.txt
12 tecmint.txt
```

مثال: لو أردنا عرض عدد الكلمات في الملف tecmint.txt نكتب:

```
[root@tecmint ~]# wc -w tecmint.txt
16 tecmint.txt
```

مثال: لو أردنا عرض عدد المحارف في الملف tecmint.txt نكتب:

```
[root@tecmint ~]# wc -m tecmint.txt
112 tecmint.txt
```

مثال: لو أردنا طول أطول سطر في الملف tecmint.txt نكتب:

```
[root@tecmint ~]# wc -L tecmint.txt
16 tecmint.txt
```

ملاحظة: هنا لو أردنا أن نعرض معلومات عددية عن ناتج تنفيذ التعليمة ls بدلاً من عرض معلومات محتوى الملف، نحدد خرج تنفيذ التعليمة ls ليكون دخلاً للتعليمة wc كما يلي:

```
[me@linuxbox ~]$ ls /bin /usr/bin | sort | uniq | wc -l
2728
```

6) **الأمر grep :**

يستخدم هذا الأمر لعرض أسطر تحقق شرطاً معيناً ، حيث أنه يقوم بمطابقة أنماط نصية داخل الملفات.

الصيغة العامة : تعرض الأسطر التي تحقق شرط البحث	grep search_text filename
تعرض فقط عدد الأسطر المطابقة لشرط البحث	grep –c search_text filename
تعرض الأسطر المطابقة للشرط مع عرض رقم كل سطر منها	grep –n search_text filename
يهمل حالة الأحرف في المقارنة حيث أن المقارنة افتراضياً حساسة لحالة الأحرف	grep –i search_text filename

يقوم هذا الأمر بمطابقة النمط search_text مع أسطر الملف filename ثم يعرض الأسطر التي تطابق النمط فقط.

مثال: لو أردنا عرض الأسطر التي تحتوي كلمة Monday فقط من الملف info.txt الموجود في المجلد home نكتب:

grep Monday /home/info.txt

ملاحظة: لو أردنا عرض الملفات والمجلدات التي تحتوي كلمة zip فقط من محتويات المسار /usr/bin والمسار /bin (بعد ترتيبهما وحذف التكرارات) نقوم بما يلي :

```
[me@linuxbox ~]$ ls /bin /usr/bin | sort | uniq | grep zip
bunzip2
bzip2
gunzip
gzip
unzip
zip
zipcloak
zipgrep
zipinfo
zipnote
zipsplit
```