

File names.txt

searching for strings:

```
grep Raman names.txt
cat names.txt | grep Raman
grep Sa names.txt
grep Anu names.txt
grep 'ai' names.txt
```

```
man dmesg
dmesg > dmesg.txt
grep VGA dmesg.txt
grep usb dmesg.txt
```

Use of dot:

```
grep 'S.n' names.txt
grep '.am' names.txt
grep '.am.' names.txt
grep 'M.' names.txt
```

Use of \ to escape special meaning:

```
grep '\.' names.txt
grep ' .\.' names.txt
```

Use of ^ for start of line:

```
grep '^M' names.txt
grep '^E' names.txt
```

Use of \$ for end of line:

```
grep 'm$' names.txt
grep 'n$' names.txt
```

Use of \b for word boundary:

```
grep '.n\b' names.txt
```

Use of [] for character set:

```
grep 'M[ME]' names.txt
grep 'E[ED]' names.txt
grep 'S.*[mn]' names.txt
```

```
grep '[aeiou][aeiou]' names.txt
```

Use of [] for character range:

```
grep 'B90[3-6]' names.txt
```

```
grep '[M-Z][aeiou]' names.txt
```

Use of ^ inside [] for character range to avoid:

```
grep 'B90[1-3]' names.txt
```

```
grep 'B90[^1-3]' names.txt
```

```
grep '[M-Z][aeiou]' names.txt
```

Use of \{n,m} for a character:

```
grep 'M\{2\}' names.txt
```

```
grep 'M\{1,2\}' names.txt
```

```
grep 'E\{1,2\}' names.txt
```

Use of \(\) for grouping of characters:

```
grep '\(ma\).*\1' names.txt
```

```
grep '\(a\).*\1' names.txt
```

```
grep '\(a\)\{2\}' names.txt
```

```
grep '\(a\)\{3\}' names.txt
```

Use of egrep or grep -E :

```
egrep '^M+' names.txt
```

```
egrep '^E+' names.txt
```

```
egrep '(ma)+' names.txt
```

```
egrep '(ma)*' names.txt
```

```
egrep '(ED|ME)' names.txt
```

```
egrep '(an|am)' names.txt
```

```
egrep '(an|am)$' names.txt
```

List four letter package names in the system :

```
dpkg-query -W -f'${Section} ${binary:Package}\n' | egrep  
' .{4}$'
```

List package names that start with g :

```
dpkg-query -W -f'${Section} ${binary:Package}\n' | egrep  
' g.*$'
```

Count package names that start with g :

```
dpkg-query -W -f'${Section} ${binary:Package}\n' | egrep  
' g.*$' | wc -l
```

List package names that start with k :

```
dpkg-query -W -f'${Section} ${binary:Package}\n' | egrep  
' k.*$'
```

List package names that belong to math category :

```
dpkg-query -W -f'${Section} ${binary:Package}\n' | egrep  
'^math'
```

cat chartypes.txt for character set examples

grep '[:alpha:]' chartypes.txt

alphabetical chars:

grep '^[:alpha:]' chartypes.txt

grep '[:alpha:]\$' chartypes.txt

alphanumeric chars:

grep '^[:alnum:]' chartypes.txt

grep '[:alnum:]\$' chartypes.txt

decimal digits:

grep '^[:digit:]' chartypes.txt

grep '[:digit:]\$' chartypes.txt

control chars:

grep '^[:cntrl:]' chartypes.txt

grep '[:cntrl:]\$' chartypes.txt

punctuation symbols:

grep '^[:punct:]' chartypes.txt

grep '[:punct:]\$' chartypes.txt

lower case char:

grep '^[:lower:]' chartypes.txt

grep '[:lower:]\$' chartypes.txt

upper case char:

grep '^[:upper:]' chartypes.txt

grep '[:upper:]\$' chartypes.txt

```
printable char:
grep '^[:print:]' chartypes.txt
grep '[:print:]+$' chartypes.txt
```

```
space or tab:
grep '^[:blank:]' chartypes.txt
grep '[:blank:]+$' chartypes.txt
```

```
space : white space char
grep '^[:space:]' chartypes.txt
grep '[:space:]+$' chartypes.txt
```

```
graph : non-space char
grep '^[:graph:]' chartypes.txt
grep '[:graph:]+$' chartypes.txt
```

```
skip all empty lines
```

```
egrep -v '^$' chartypes.txt
```

Picking up patterns from a text file:

```
cat patterns.txt
```

```
egrep '[:digit:]{12}' patterns.txt
```

```
egrep '[:digit:]{10}' patterns.txt
```

Use \b for word boundary:

```
egrep '\b[:digit:]{10}\b' patterns.txt
```

```
egrep '[:digit:]{6}' patterns.txt
```

```
egrep '\b[:digit:]{6}\b' patterns.txt
```

```
egrep '[:alpha:]{2}[:digit:]{2}[:alpha:][:digit:]{3}' patterns.txt
```

```
egrep '\b[:alnum:]+\.[[:alnum:]]+\b' patterns.txt
```

```
cat fields.txt
```

Using characters:

```
cat fields.txt | cut -c 1-4  
cat fields.txt | cut -c 6-10
```

Using space as delimiter:

```
cat fields.txt | cut -d " " -f 1  
cat fields.txt | cut -d " " -f 1-2  
cat fields.txt | cut -d ";" -f 1  
cat fields.txt | cut -d ";" -f 2  
cat fields.txt | cut -d "," -f 1  
cat fields.txt | cut -d "," -f 2
```

Combining cut to change delimiter:

```
cat fields.txt | cut -d ";" -f 2 | cut -d "," -f 1
```

Combining cut, head and tail :

```
cat fields.txt | cut -d ";" -f 2 | cut -d "," -f 1 | head  
-n 2 | tail -n 1
```