

Select delimited fields

```
$ head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
$ cut -d: -f 1 /etc/passwd | head -5 # Output field 1
root
daemon
bin
sys
sync
$ cut -d: -f 3-4 /etc/passwd | head -5 # Output fields 3-4
0:0
1:1
2:2
3:3
4:65534
$
```

Select fixed-length fields

```
$ grep -vi '^c' spice/acans.f | # Remove comment lines
> cut -c 2-6 | # Output label field
> tail # Output last ten labels

910

920

1000
2000

$
```

Conditional selection with awk

```
$ awk '/bash/' /etc/passwd # Output lines containing "bash"
root:x:0:0:root:/root:/bin/bash
dds:x:1001:1001:Diomidis Spinellis,,:/home/dds:/bin/bash
$ awk -F: '$3 > 1000' /etc/passwd # Lines where field 3 > 1000
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
dds:x:1001:1001:Diomidis Spinellis,,:/home/dds:/bin/bash
$ awk -F: '{print $1}' /etc/passwd | head -5 # Output field 1
root
daemon
bin
sys
sync
$ awk '!/^#/ {print $1}' /etc/services | head # Combine predicate and action

tcpmux
echo
echo
discard
discard
sysstat
daytime
daytime
netstat
$
```

RE selection with sed

```
$ cd /usr/src/linux/kernel # Linux kernel source code directory
$ sed -n '/#include *["<"]\[^\>]*\.\.\/\1/p' *.c | # Output included file names
> head
linux/mm.h
linux/slab.h
linux/acct.h
linux/capability.h
linux/file.h
linux/tty.h
linux/security.h
linux/vfs.h
linux/jiffies.h
linux/times.h
$
```

Select which lines to process

```
$ cd /usr/share/dict # Output lines from lines 1000 to 1005
$ sed -n 1000,1005p words
Ashikaga
Ashikaga's
Ashkenazim
Ashkhabad
Ashkhabad's
AshLee
$ cd /usr/src/linux/kernel/printk
$ sed -n '/^enum log_flags/,/^};$/p' printk.c # Output log_flags definition
enum log_flags {
    LOG_NOCONS      = 1,    /* already flushed, do not print to console */
    LOG_NEWLINE     = 2,    /* text ended with a newline */
    LOG_PREFIX      = 4,    /* text started with a prefix */
    LOG_CONT        = 8,    /* text is a fragment of a continuation line */
};
```

```
};  
$
```

Select JSON data

```
$ curl -q 'http://api.geonames.org/citiesJSON?north=37&south=38&east=24&west=23&lang=en&username=demo&maxRows=1' >result.json  
$ cat result.json  
{  
  "geonames": [  
    {  
      "lng": 23.7162208557129,  
      "geonameId": 264371,  
      "countrycode": "GR",  
      "name": "Athens",  
      "fclName": "city, village, ...",  
      "toponymName": ""  
    }  
  ]  
}  
$ jq -r '.geonames[0].name, .geonames[0].countrycode' result.json  
Athens  
GR  
$
```

Select XML data

```
$ curl -q 'http://api.geonames.org/cities?north=51&south=52&east=0&west=1&lang=en&username=demo&maxRows=1' >result.xml  
$ cat result.xml  
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<geonames>  
  <geoname>  
    <toponymName>London</toponymName>  
    <name>London</name>  
    <lat>51.50853</lat>  
    <lng>-0.12574</lng>  
    <geonameId>2643743</geonameId>  
    <countryCode>GB</countryCode>  
    <countryName>United Kingdom</countryName>  
    <fcl>P</fcl>  
    <fcode>PPLC</fcode>  
  </geoname>  
</geonames>  
$ xmlstarlet sel -t -c /geonames/geoname/name result.xml  
<name>London</name>  
$
```