

More tr examples

`tr '&' '#'` translate ampersands to hash

`tr -s '\t'` squeeze consecutive tabs to one tab



More tr examples

```
$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
128.186.120.8     sophie.cs.fsu.edu
127.0.0.1         a.as-us.falkag.net
127.0.0.1         clk.atdmt.com
$ tr -s '\t' < /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
128.186.120.8     sophie.cs.fsu.edu
127.0.0.1         a.as-us.falkag.net
127.0.0.1         clk.atdmt.com
```



More tr examples

```
tr -d '\015'    delete carriage returns from a DOS file
```



`basename` lets you remove leading directory strings. It can also remove suffixes simply by specifying the suffix as a second argument.

```
$ basename `pwd`  
2006-Fall  
$ var1=/etc/inetd.conf  
$ basename $var1 .conf  
inetd
```



`dirname` does the opposite function of `basename`: it returns the leading path components from a directory name.

```
$ echo `pwd`  
/mnt-tmp/Lexar/fsucs/cop-4342/2006-Fall  
$ dirname `pwd`  
/mnt-tmp/Lexar/fsucs/cop-4342  
$ dirname 05-shells4.tex  
.  
$ dirname `pwd`/xyz  
/mnt-tmp/Lexar/fsucs/cop-4342/2006-Fall
```



For all of the files listed, `sort` will sort the concatenated lines of those files to `stdout`. The most useful options are `-f`, which means to fold case, `-n` to sort numerically rather alphabetically, `-u` to remove duplicates (“u” is short for “unique”), and `-r` to reverse the order of the sort.

You can specify particular fields to sort by specifying a field separator (whitespace is the default) with the `-t` option, and then using `-k` to specify particular fields.



sort examples

```
$ sort /etc/passwd
adm:x:3:4:adm:/var/adm:/sbin/nologin
amanda:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
canna:x:39:39:Canna Service User:/var/lib/canna:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
desktop:x:80:80:desktop:/var/lib/menu/kde:/sbin/nologin
```



sort examples

```
$ sort -r /etc/passwd
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
wnn:x:49:49:Wnn Input Server:/var/lib/wnn:/sbin/nologin
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin
vmail:x:502:502::/home/vmail:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
user1:x:505:505::/home/user1:/bin/bash
test:x:503:503::/home/test:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```



sort examples

```
$ sort -k3,3n -t: /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```



sort examples

```
$ sort -k4,4n -k3,3n -t: /etc/passwd
root:x:0:0:root:/root:/bin/bash
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
operator:x:11:0:operator:/root:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
```



- There are a lot of great packages out there, such as `graphviz`. A handy one is `groff`, a derivative of the ancient `troff` and `nroff` families. (“roff” comes from “runoff”; `man` pages are traditionally written in `nroff` format.)
- You can use `gtbl` with `groff` to quickly make nice PostScript tables.

```
gtbl some.tr | groff > /tmp/some.ps
```



groff and gtbl

```
.sp 10      # skip 10 lines
.ps 14      # point size 14 pt
.TS        # table start
center box tab(/); # center the table in the page, put it in a box,
c c c c    # center the first line
r r r r .  # right justify the rest
.sp .2v    # skip down 2/10s of a line
Last / First / Age / Zipcode    # column headers
.sp .1v    # skip down 1/10 of a line
_          # horizontal rule
_          # horizontal rule
.sp .3v    # skip down 3/10s of a line
Gordon/Flash/91/91191    # record one
.sp .2v    # skip down
Jones/Carol/20/32399    # record two
.sp .2v    # skip down
Miller/Bob/23/32499    # record three
.sp .2v    # skip down
Yagi/Akihito/22/32111    # record four
.sp .1v    # skip down
.TE        # table end
```



Another great little utility is `fmt` which lets you quickly reformat a document.

You can use `-w` to control the width. `fmt` also prefers to see two spaces after a question mark, period, or exclamation point to indicate the end of a sentence.



```
$ cat lincoln.txt
```

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.



fmt example

```
$ fmt lincoln.txt
```

```
Four score and seven years ago our fathers brought forth on this  
continent, a new nation, conceived in Liberty, and dedicated to the  
proposition that all men are created equal.
```

```
Now we are engaged in a great civil war, testing whether that nation,  
or any nation so conceived and so dedicated, can long endure. We are m  
on a great battle-field of that war. We have come to dedicate a portio  
of that field, as a final resting place for those who here gave their  
lives that that nation might live. It is altogether fitting and proper  
that we should do this.
```



fmt example

```
$ fmt -w 20 lincoln.txt
Four score and
seven years ago our
fathers brought
forth on this
continent, a new
nation, conceived
in Liberty, and
dedicated to the
proposition that
all men are created
equal.
```



- `cut` allows you to extract columnar portions of a file. The columns can be specified either by a delimiter (the default delimiter is the tab character.)
- You can specify a delimiter with the `-d` option.
- You must specify either at least one field number with `-f`, a byte number with `-b`, or a character number with `-c`. With ordinary ASCII text, `-b` and `-c` mean the same thing, but if we ever get multi-byte characters handled correctly, it shouldn't.



cut examples

```
$ cut -c 1 /etc/hosts
#
#
1
1
1
1
```



cut examples

```
$ cut -b 1 /etc/hosts
#
#
1
1
1
1
```



cut examples

```
$ cut -f1 /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1
128.186.120.8
127.0.0.1
127.0.0.1
```



cut examples

```
$ cut -c1-10 /etc/hosts
# Do not r
# that req
127.0.0.1
128.186.12
127.0.0.1
127.0.0.1
```



cut examples

```
$ cut -d: -f1,5 /etc/passwd
netdump:Network Crash Dump user
sshd:Privilege-separated SSH
rpc:Portmapper RPC user
rpcuser:RPC Service User
nfsnobody:Anonymous NFS User
```



lets you put two or more files together as columns. By default, the columns will be joined with a tab character, but you can use the `-d` option to specify a different delimiter.



paste example

```
prompt% cut -f1 /etc/hosts > /tmp/f1
prompt% cut -d: -f5 /etc/passwd /tmp/f2
prompt% paste -d: /tmp/f1 /tmp/f2
# Do not remove the following line, or various programs:root
# that require network functionality will fail.:bin
127.0.0.1:daemon
128.186.120.8:adm
127.0.0.1:lp
127.0.0.1:sync
```



- These programs, as mentioned before, allow you to excerpt the initial or the final lines of a file.
- Used in combination, you can isolate an arbitrary range of lines.
- You can also use the `-f` option with `tail` to monitor a file for changes.
- By default, if you specify multiple files, you get a nice little header to distinguish them.



head and tail examples

```
head /etc/passwd      # print the first 10 lines of passwd
tail -20 /etc/passwd  # print the last 20 lines of passwd
head -15 /etc/passwd | tail -5 # print lines 10 - 15 of passwd
tail -f /var/log/messages # monitor the log ``messages`` file
```



Chapter 34 of UPT has a good section on `sed`.
`sed` is a “stream editor.” It can edit files in place.
You can specify multiple `sed` scripts with `-e`.



sed examples

```
$ sed "s/1/9/" < /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
927.0.0.1          localhost.localdomain localhost
928.186.120.8     sophie.cs.fsu.edu
927.0.0.1         a.as-us.falkag.net
927.0.0.1         clk.atdmt.com
```



sed examples

```
$ sed -e "s/l/9/" -e "s/a/A/g" < /etc/hosts
# Do not remove the following line, or vArious progrAms
# thAt require network functionAlity will fAil.
927.0.0.1          locAlhost.locAldomAin locAlhost
928.186.120.8     sophie.cs.fsu.edu
927.0.0.1         A.As-us.fAlkAg.net
927.0.0.1         clk.Atdmt.com
```

