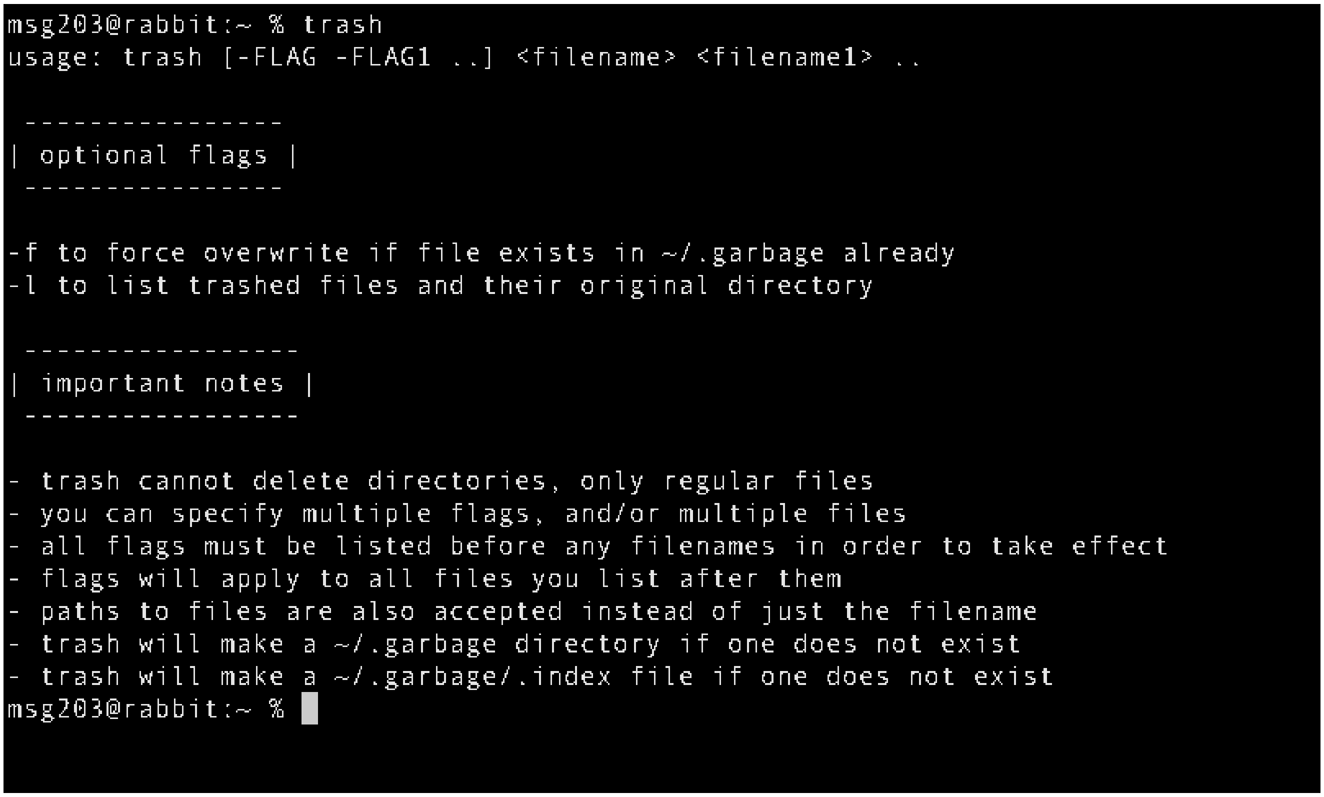
ECE 322 FALL 2021

HW #1 (trash/untrash scripts)

Myles Greene - Beaupre

trash

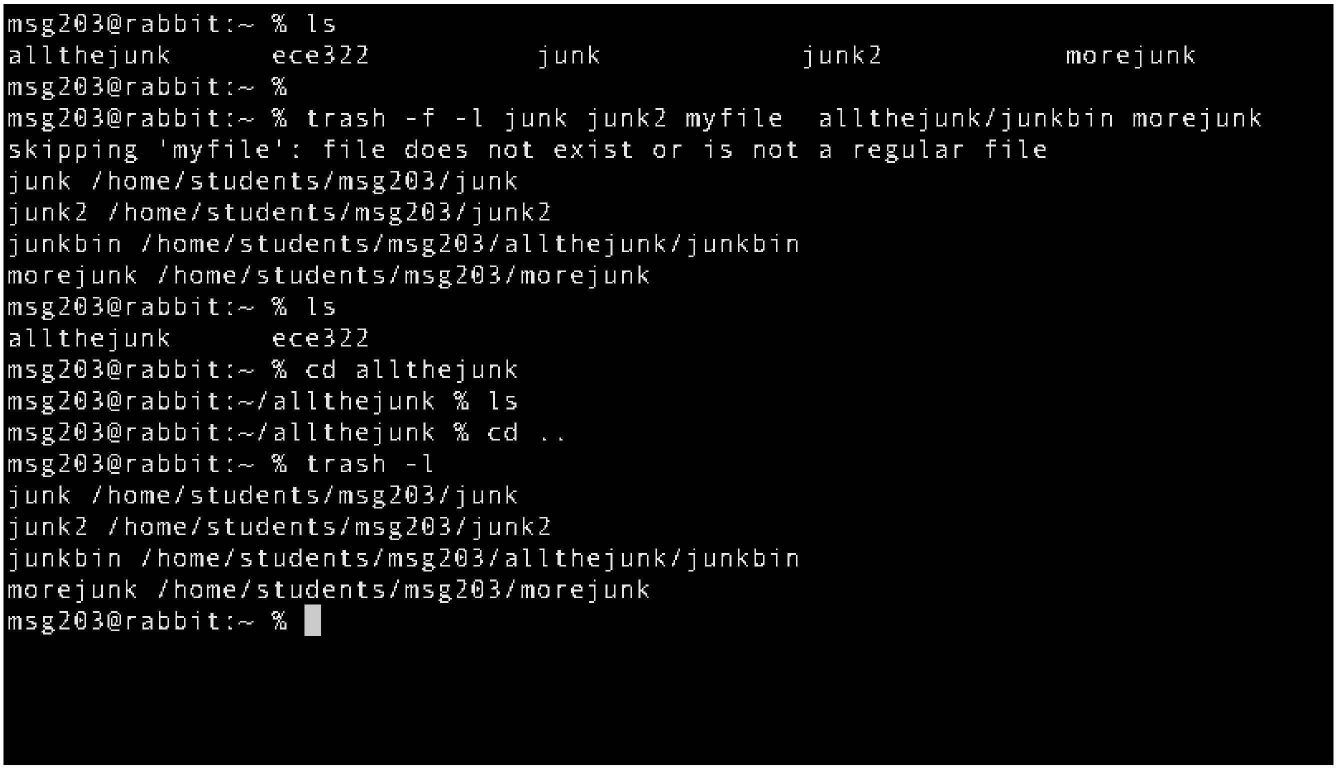
running simply trash (or home/students/msg203/ece322/trash/trash.csh without the alias) will give information on usage, flags, and the way it works in general:



```
msg203@rabbit:~ % trash
usage: trash [-FLAG -FLAG1 ..] <filename> <filename1> ..

 ----------------
| optional flags |
 ----------------

-f to force overwrite if file exists in ~/.garbage already
-l to list trashed files and their original directory

 -----------------
| important notes |
 -----------------

- trash cannot delete directories, only regular files
- you can specify multiple flags, and/or multiple files
- all flags must be listed before any filenames in order to take effect
- flags will apply to all files you list after them
- paths to files are also accepted instead of just the filename
- trash will make a ~/.garbage directory if one does not exist
- trash will make a ~/.garbage/.index file if one does not exist
msg203@rabbit:~ %
```

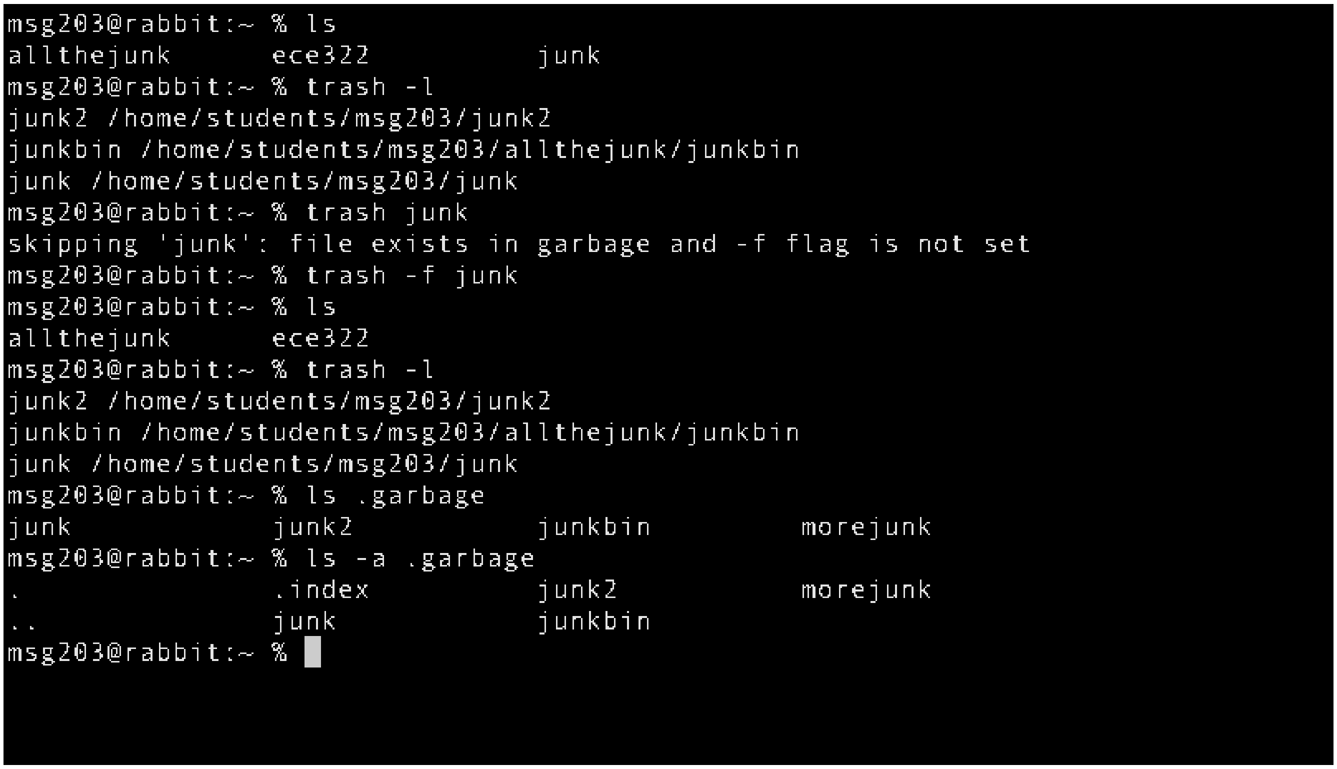
you can specify as many flags and as many filenames as you want, and they will all be executed so long as you put all the flags first before you put any filenames

ex) you can delete several files, force overwrite for all of them, and then print out the contents of the garbage all in one command; also, if you make a typo when writing a filename or you try to delete a file that doesn’t exist, trash.csh will simply skip it and continue running



```
msg203@rabbit:~ % ls
allthejunk      ece322          junk            junk2           morejunk
msg203@rabbit:~ %
msg203@rabbit:~ % trash -f -l junk junk2 myfile  allthejunk/junkbin morejunk
skipping 'myfile': file does not exist or is not a regular file
junk /home/students/msg203/junk
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
morejunk /home/students/msg203/morejunk
msg203@rabbit:~ % ls
allthejunk      ece322
msg203@rabbit:~ % cd allthejunk
msg203@rabbit:~/allthejunk % ls
msg203@rabbit:~/allthejunk % cd ..
msg203@rabbit:~ % trash -l
junk /home/students/msg203/junk
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
morejunk /home/students/msg203/morejunk
msg203@rabbit:~ %
```

and if you do not specify the force flag, trash.csh will not overwrite files with the same name in the garbage directory:

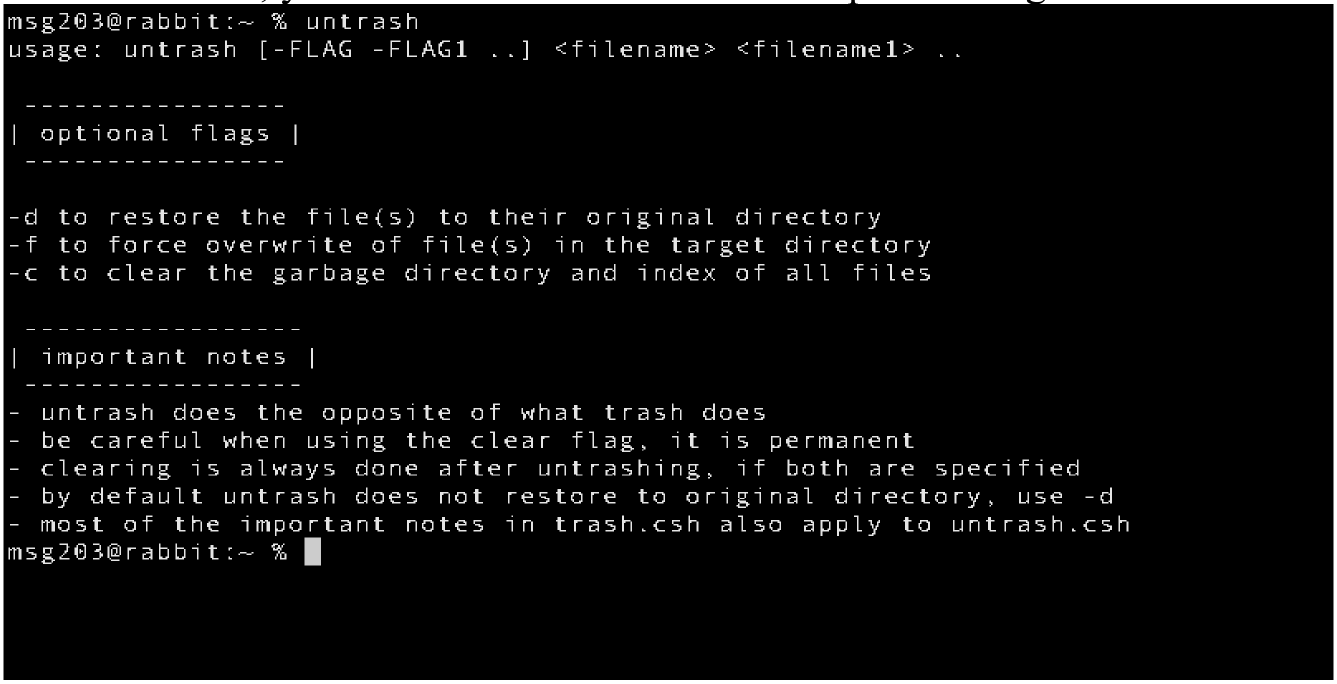


```
msg203@rabbit:~ % ls
allthejunk      ece322          junk
msg203@rabbit:~ % trash -l
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
junk /home/students/msg203/junk
msg203@rabbit:~ % trash junk
skipping 'junk': file exists in garbage and -f flag is not set
msg203@rabbit:~ % trash -f junk
msg203@rabbit:~ % ls
allthejunk      ece322
msg203@rabbit:~ % trash -l
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
junk /home/students/msg203/junk
msg203@rabbit:~ % ls .garbage
junk            junk2           junkbin         morejunk
msg203@rabbit:~ % ls -a .garbage
.               .index          junk2           morejunk
..              junk            junkbin
msg203@rabbit:~ %
```

trash creates .garbage and .index automatically if either doesn’t exist

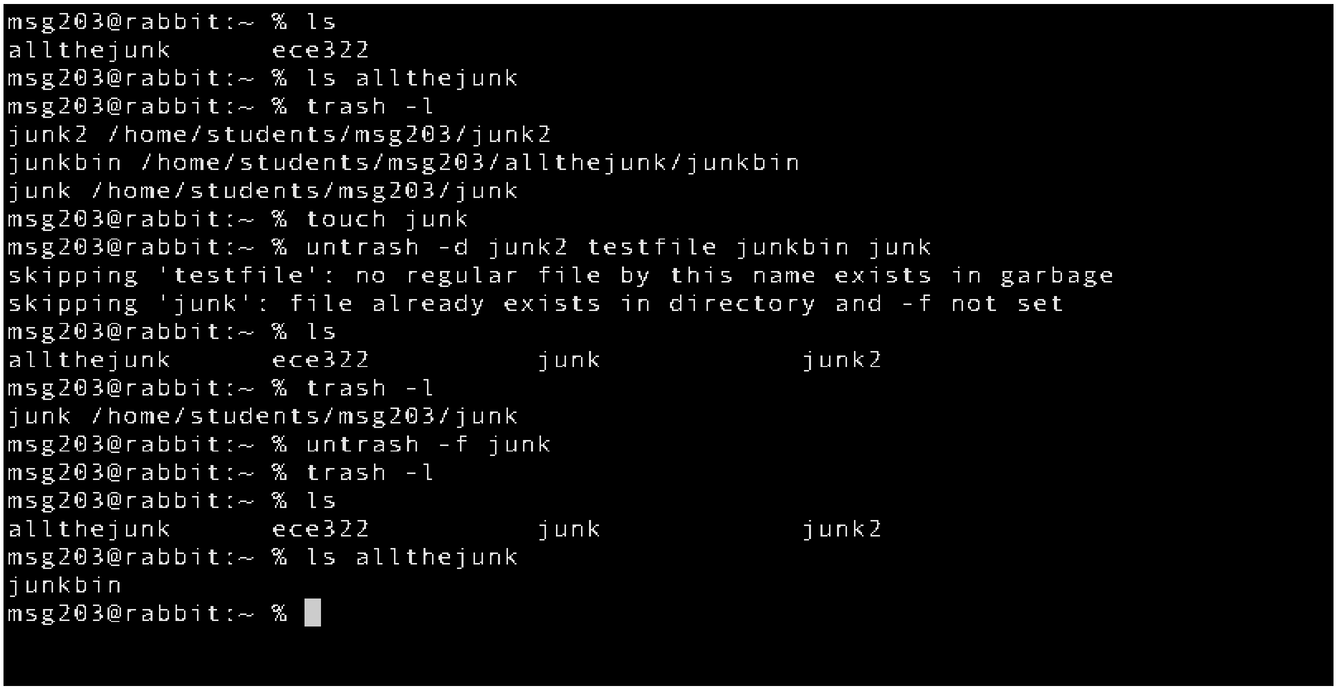
untrash

similar to trash, you can also run untrash with no params to get info:



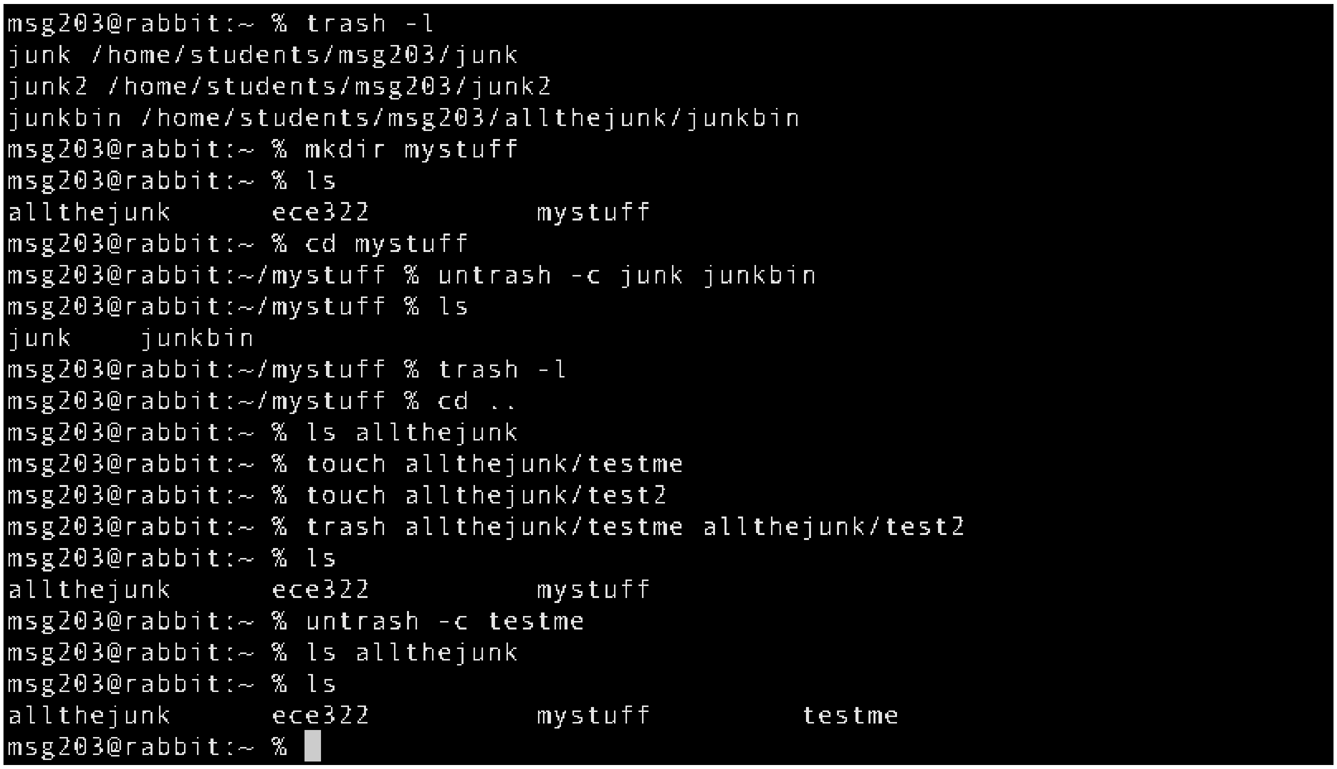
```
msg203@rabbit:~ % untrash
usage: untrash [-FLAG -FLAG1 ..] <filename> <filename1> ..

 ----------------
| optional flags |
 ----------------

-d to restore the file(s) to their original directory
-f to force overwrite of file(s) in the target directory
-c to clear the garbage directory and index of all files

 -----------------
| important notes |
 -----------------
- untrash does the opposite of what trash does
- be careful when using the clear flag, it is permanent
- clearing is always done after untrashing, if both are specified
- by default untrash does not restore to original directory, use -d
- most of the important notes in trash.csh also apply to untrash.csh
msg203@rabbit:~ % 
```

everything works the same as with trash.csh, you can do multiple flags and multiple files all at once, and it has the same skipping mechanism for non-existent files and for duplicates when -f is not specified



```
msg203@rabbit:~ % ls
allthejunk      ece322
msg203@rabbit:~ % ls allthejunk
msg203@rabbit:~ % trash -l
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
junk /home/students/msg203/junk
msg203@rabbit:~ % touch junk
msg203@rabbit:~ % untrash -d junk2 testfile junkbin junk
skipping 'testfile': no regular file by this name exists in garbage
skipping 'junk': file already exists in directory and -f not set
msg203@rabbit:~ % ls
allthejunk      ece322          junk            junk2
msg203@rabbit:~ % trash -l
junk /home/students/msg203/junk
msg203@rabbit:~ % untrash -f junk
msg203@rabbit:~ % trash -l
msg203@rabbit:~ % ls
allthejunk      ece322          junk            junk2
msg203@rabbit:~ % ls allthejunk
junkbin
msg203@rabbit:~ % 
```

if you specify the clear flag and untrash files at the same time, the garbage will be emptied after the specified files are restored (in this example I do not specify the -d flag so I am able to untrash the files to my current directory)



```
msg203@rabbit:~ % trash -l
junk /home/students/msg203/junk
junk2 /home/students/msg203/junk2
junkbin /home/students/msg203/allthejunk/junkbin
msg203@rabbit:~ % mkdir mystuff
msg203@rabbit:~ % ls
allthejunk      ece322          mystuff
msg203@rabbit:~ % cd mystuff
msg203@rabbit:~/mystuff % untrash -c junk junkbin
msg203@rabbit:~/mystuff % ls
junk    junkbin
msg203@rabbit:~/mystuff % trash -l
msg203@rabbit:~/mystuff % cd ..
msg203@rabbit:~ % ls allthejunk
msg203@rabbit:~ % touch allthejunk/testme
msg203@rabbit:~ % touch allthejunk/test2
msg203@rabbit:~ % trash allthejunk/testme allthejunk/test2
msg203@rabbit:~ % ls
allthejunk      ece322          mystuff
msg203@rabbit:~ % untrash -c testme
msg203@rabbit:~ % ls allthejunk
msg203@rabbit:~ % ls
allthejunk      ece322          mystuff         testme
msg203@rabbit:~ %
```

<u>source code</u>

both scripts have a foreach that loops over all the elements of $argv, which first checks for flags, sets variables, and then works on files accordingly

trash.csh:

```
#!/bin/csh

if (${#argv} < 1) then
  echo "usage: trash [-FLAG -FLAG1 ..] <filename> <filename1> .."
  echo ""
  echo " ----------------"
  echo "| optional flags |"
  echo " ----------------"
  echo ""
  echo "-f to force overwrite if file exists in ~/.garbage already"
  echo "-l to list trashed files and their original directory"
  echo ""
  echo " ----------------"
  echo "| important notes |"
  echo " ----------------"
  echo ""
  echo "- trash cannot delete directories, only regular files"
  echo "- you can specify multiple flags, and/or multiple files"
  echo "- all flags must be listed before any filenames in order to take effect"
  echo "- flags will apply to all files you list after them"
  echo "- paths to files are also accepted instead of just the filename"
  echo "- trash will make a ~/.garbage directory if one does not exist"
  echo "- trash will make a ~/.garbage/.index file if one does not exist"
  exit -1
endif

set gardir = ~/.garbage
set indexfile = ~/.garbage/.index

if (! -d $gardir) then
  mkdir $gardir
  touch $indexfile
else if (! -e $indexfile) then
  touch $indexfile
endif

foreach i ($argv)
```

```
  if ("$i" =~ "-f") then
    set force
  else if ("$i" =~ "-l") then
    set list
  else
    if (! -f $i) then
      echo "skipping '$i': file does not exist or is not a 
regular file"
    else
      set filename = `basename $i`
      if (-e $gardir/$filename && ! $?force) then
        echo "skipping '$i': file exists in garbage and -f 
flag is not set"
      else
        set filepath = `readlink -f $i`
    cp $i $gardir
    rm $i
    sed -i "" '/'$filename'$/d' $indexfile
    echo $filename $filepath >> $indexfile
      endif
    endif
  endif
end

if ($?list) then
  cat $indexfile
endif

exit 0
```

untrash.csh:

```
#!/bin/csh

if (${#argv} < 1) then
  echo "usage: untrash [-FLAG -FLAG1 ..] <filename> 
<filename1> .."
  echo ""
  echo " ----------------"
  echo "| optional flags |"
  echo " ----------------"
  echo ""
  echo "-d to restore the file(s) to their original 
directory"
  echo "-f to force overwrite of file(s) in the target 
directory"
  echo "-c to clear the garbage directory and index of all 
files"
  echo ""
  echo " -----------------"
  echo "| important notes |"
  echo " -----------------"
  echo "- untrash does the opposite of what trash does"
  echo "- be careful when using the clear flag, it is 
permanent"
  echo "- clearing is always done after untrashing, if both 
are specified"
  echo "- by default untrash does not restore to original 
directory, use -d"
  echo "- most of the important notes in trash.csh also 
apply to untrash.csh"
  exit -1
endif

set gardir = ~/.garbage
set indexfile = ~/.garbage/.index

if (! -d $gardir) then
  echo "garbage directory does not exist, please trash 
something first"
  exit -1
else if (! -e $indexfile) then
  echo "index file is missing from garbage directory, this 
will cause bugs"
  exit -1
endif
```

```
foreach i ($argv)
  if ("$i" =~ "-f") then
    set force
  else if ("$i" =~ "-d") then
    set origdir
  else if ("$i" =~ "-c") then
    set clear
  else
    if (! -f $gardir/$i) then
      echo "skipping '$i': no regular file by this name 
exists in garbage"
    else
      if ($?origdir) then
        set dir = `cat $indexfile | cut -f 2 -d ' ' | grep 
''$i'$'`
      else
    set dir = $PWD/$i
      endif
      if (-e $dir && ! $?force) then
        echo "skipping '$i': file already exists in 
directory and -f not set"
      else
        cp $gardir/$i $dir
    rm $gardir/$i
    sed -i "" '/'$i'$/d' $indexfile
      endif
    endif
  endif
end

if ($?clear) then
  rm $gardir/*
  rm $indexfile
  touch $indexfile
endif

exit 0
```