Slide 1

Administrivia

- Information about projects on the Web. Due dates are November 22 (proposals), date of final (everything else).
- There will also be one more homework, also due the day of the final.

Slide 2

“Programming” Problems

- Today’s class — for a couple of semi-real-world problems, think of as many ways as we can to solve them using tools presented so far.

Slide 3

Mass File Rename

- How many ways can we think of to do a “mass rename” of files? e.g., rename all `.cc` files to `.cpp`?
- (A really good solution will do something reasonable if aren’t any `.cc` files, and will work even if file names contain spaces.)
- Linux has a `rename` command — exactly what we want, but not available on all UNIX systems, and — are there other ways?

Slide 4

Mass File Rename, Continued

- Another way:

```
for file in ‘ls *.cc‘
do
  mv $file ‘basename $file .cc‘.cpp
done
```

  (Or could use `sed` to make new file name.)

- A problem — the above fails if filename contains spaces. It also doesn’t do the right thing if there aren’t any files. How to fix this?

Mass File Rename, Continued

- One way is to use a `while` loop rather than a `for`:

```
ls *.cc | while read file
do
  mv "$file" "‘basename \"$file\" .cc‘.cpp"
done
```

Slide 5

Finding Broken Links

- How many ways can we think of to find “broken” (or “stale”) symbolic links?
- Linux has a `symlinks` command too. Other ways?

Slide 6

Slide 7

Finding Broken Links, Continued

- Another way would be to combine `find` and `readlink`, e.g.:

```
for link in ‘find . -type l‘
do
  file=‘readlink $link‘
  if [ ! -e $file ]
  then
    echo broken link $link
  fi
done
```

- (As a student pointed out in class, we could also use `readlink` with the `-e` flag and check either the result (blank) or the return code (nonzero).
- But this only works for links in the current directory — for links in subdirectory, test for link target’s existence is wrong. How to fix this?

Slide 8

Finding Broken Links, Continued

- The following seems to work for both Linux and OS X:

```
find . -type l | while read link
do
  dir=‘dirname $link‘
  name=‘basename $link‘
  here=‘pwd‘
  cd $dir
  target=‘readlink $name‘
  if [ ! -e $target ]
  then
    echo "broken link $dir/$name"
  fi
  cd $here
done
```

(though it needs tweaking to deal with spaces in filenames).

Slide 9

More Real-World(?) Examples — Locally-Written Scripts

- `p`
- `find-big-files`, `sorted-disk-usage`
- `up`, `atlas`, etc.
- `logon`
- (These are in `/usr/local/bin`.)

Slide 10

Minute Essay

- None — sign in.