

### Administrivia

- Reminder: Homework 5 was officially due last week, but no late penalty if turned in today. Homework 6 due Wednesday.
- Today's topic may need explanation — address some questions and things we skipped over in previous lectures, allow catching up with homework.

Slide 1

### UNIX File Links

- UNIX files identified by paths in a basically hierarchical scheme. Directories are basically a special type of file.
- Usually files/directories form a tree (with files as leaf nodes). Links allow the creation of more general graphs (ideally acyclic ones!).

Slide 2

### Hard Links

- Traditional UNIX filesystems have an “inode” for each file — something that contains information about the file (owner, permissions, etc.) plus its location on disk. Directory entries point to inodes.
- “Hard links” are a way of pointing more than one directory entry to the same inode. Actual file deleted when there are no more hard links to it.

Slide 3

### Symbolic (“Soft”) Links

- “Symbolic links” are another special type of file, containing a path name.
- No check is made to ensure that the link target exists! so they can be invalid.
- `readlink` shows target of link.

Slide 4

## Arrays in bash

Slide 5

- Yes, `bash` variables can be arrays. Can be indexed or associative. See `man` page (or full manual) for details.

- Indexed arrays can be declared but don't have to be:

```
foo[0]="hello" ; foo[1]=5  
echo ${foo[1]} ; echo ${foo[*]}
```

- Associative arrays must be declared:

```
declare -A bar  
bar[a]="hello" ; bar[z]=5  
echo ${bar[z]} ; echo ${bar[*]}
```

## Interesting(?) Uses of Environment Variables

Slide 6

- Environment variables can contain almost anything. One thing I use them for is frequently-visited directories. (Symbolic links would work for that too.)
- "Current directory" is a useful concept that sometimes makes it easier to specify pathnames (contrast with GUI file-chooser menu). What if you want to have two of them, though? Could define a shell function or alias that sets an environment variable, and reference that. I have in my `.bashrc`:

```
alias qqdir='QQ=`pwd`;export QQ'
```

## Command-Line Editing

Slide 7

- `bash` provides support for some fairly sophisticated editing of command lines. You know about some features already (command history, tab completion for filenames).
- Support for other editing functions based on `emacs` key bindings. One I particularly like: `control-w` deletes the last “word” on the line, `control-y` gets it back. Makes it easier to, e.g., give a file a new name very similar to its old name.

## Minute Essay

Slide 8

- It's the middle of the semester, so probably a good time to ask: How is the pace of the course for you? level of difficulty?