Unix Commands

Set 1

Online Documentation

Getting help!

- **man** [ -a ] [ -k *keyword* ] [ -s n ] *word*
  - -a All (search all sections for word)
  - -k Keyword (i.e., searches all man-pages for *keyword*)
  - -s Search for word in section number *n* (digit, digit & letter, letter)

- **Man pages**
  - [ ] Don’t type—denotes optional parts
  - | OR—choose one or the other but not both
  - Manual sections
    - 1 commands / utilities
    - 2 system calls and error numbers
    - 3 library functions
    - 4 file formats
# Metacharacters

Characters program-specific meaning

- Some characters are processed or interpreted specially by shells
- Metacharacters must sometimes be “hidden from the shell” by quoting with `'` or `"`:

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&gt;</code></td>
<td>redirection</td>
</tr>
<tr>
<td><code>&gt;&gt;</code></td>
<td>redirection</td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>redirection</td>
</tr>
<tr>
<td><code>*</code></td>
<td>wildcard</td>
</tr>
<tr>
<td><code>?</code></td>
<td>wildcard</td>
</tr>
<tr>
<td><code>[ ]</code></td>
<td>wildcard</td>
</tr>
<tr>
<td><code>\ </code></td>
<td>command substitution</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
</tr>
<tr>
<td><code>:</code></td>
<td>command sequence</td>
</tr>
<tr>
<td>`</td>
<td></td>
</tr>
<tr>
<td><code>&amp;&amp;</code></td>
<td>conditional execution</td>
</tr>
<tr>
<td><code>( )</code></td>
<td>group commands</td>
</tr>
<tr>
<td><code>&amp;</code></td>
<td>run in background</td>
</tr>
<tr>
<td><code>#</code></td>
<td>comment to end of line</td>
</tr>
<tr>
<td><code>$</code></td>
<td>variable substitution</td>
</tr>
<tr>
<td><code>.</code></td>
<td>suppress substitutions</td>
</tr>
<tr>
<td><code>&quot; &quot;</code></td>
<td>suppress substitutions</td>
</tr>
</tbody>
</table>

# Some Unix Basics

Basic, but necessary!

- Unix is case sensitive
  - Unix commands are almost always lower case
  - The arguments might be upper or mixed case
- The name of a Unix command must always be separated from any arguments by at least one space
  - If there are multiple arguments, they must be separated by space
  - If there is a flag or option, it must be separated from the command and arguments by space
  - The command is read & run after the enter key is pressed

```
% cp hours LastWeek
% man -s4 passwd
```
**ls**

File information

- **ls** lists file information; the `-l` option produces a long listing
  - file type: - (file), d (directory), l (symbolic link), p (pipe), s (socket)
  - protection modes: read/write/execute for user/group/others
  - number of links
  - owner
  - size in bytes
  - last modification time
  - file name

```
-rw-r--r--  1 dab    2395 Apr 28 15:23 wc.c
drwxrwxr-x  3 root   7680 Jul 21 20:42 /usr/bin
```

---

**Decoding File Permissions**

File modes

- Always three sets of three permissions
- Sets (always in this order)
  - User (owner)
  - Group
  - Others
- Permissions within each set
  - r  read
  - w  write
  - x  execute
  - -  no permission

The `chmod` Utility

**Symbolic modes**

- The *newMode* is a comma-separated list (no white space) of symbolic mode expressions.
- **Who**
  - `a` (all)
  - `u` (user or owner)
- **Operators**
  - `+` (relative: add to current mode)
  - `-` (relative: subtract from current mode)
  - `=` (absolute: set mode)
- **Examples**
  - `chmod go-rwx bigdeals secret`
  - `chmod a+x script`
  - `chmod u=rw mystuff`
  - **Modes**
  - `r` (read)
  - `w` (write)
  - `x` (execute)

The `chmod` Utility

**Absolute mode**

- *newMode* is 3-digit octal number
- Permissions are expressed with octal digits
  - One digit each for user, group, and others
  - Each digit uses one bit for each permission: read, write, and execute
- Binary and octal values of modes: `rwx`
  - `r` $\equiv 100_2 \equiv 4_8$
  - `w` $\equiv 010_2 \equiv 2_8$
  - `x` $\equiv 001_2 \equiv 1_8$
  - `-` $\equiv 000_2 \equiv 0_8$
  - `|` is the bit-wise OR; use to “switch on” bits
  - `rw` $= 100 | 010 = 110 = 4 + 2 = 6$
  - `rx` $= 100 | 001 = 101 = 4 + 1 = 5$
  - `rwx` $= 100 | 010 | 001 = 111 = 4 + 2 + 1 = 7$
- **Examples**
  - `chmod 640 bigdeals secret`
  - `chmod 755 script`
  - `chmod 600 mystuff`