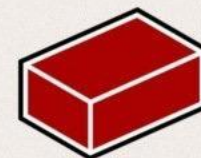


Using Redbrick

IRC Clients, Multiplexers, Web Hosting, Development

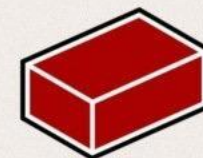


Redbrick

DCU's Networking Society

Multiplexer

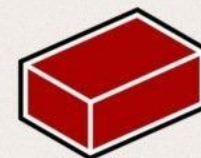
Screen or Tmux



Redbrick
DCU's Networking Society

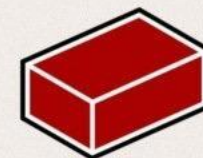
Screen

- *nano .screenrc*
- creates a status bar at the bottom of the screen
hardstatus alwayslastline
hardstatus string '%{= kg}[%{G}%H %{g}][%= %{= kw}%?%-Lw%?%{=b kR}(%{W}%n%f %\$*
- open up irc, a terminal window and a window for heys
screen -t irc 1 irssi --config=my-irssi.conf
screen -t pyg 2 ssh pygmalion
screen -t heys -L -l 0
- or the easy way
cp ~edu/.screenrc ~



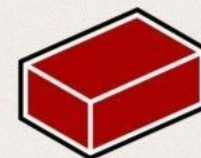
tmux

- Lighter than screen
- Better plugins
- Scriptable
- To set it up
 - `~edu/setupTmux`
 - rebinds keys `ctrl+b` > `ctrl+a`
 - makes it prettier



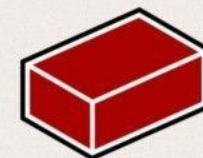
Auto Attach

- *nano ~/.zlogin*
- if you use screen
 - *screen -DR*
- if you use tmux
 - *tmux attach -d*



IRC clients

irssi or weechat



Redbrick
DCU's Networking Society

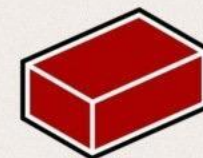
Irssi

- nano my-irssi.conf
- Auto Login

```
chatnets = {  
    RedBrick = {  
        type = "IRC"; autosendcmd = "/msg NickServ IDENTIFY lamepassword"; # put in your identify password here.  
    };  
};
```

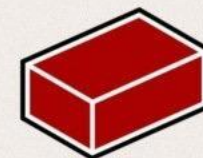
- join channels

```
channels = (  
    { name = "#lobby"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#bots"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#uno; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#wolf"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#countdown"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#CardsAgainstHumanity"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#helpdesk"; chatnet = "RedBrick"; autojoin = "yes"; }  
    { name = "#intersocs"; chatnet = "RedBrick"; autojoin = "yes"; }  
);
```



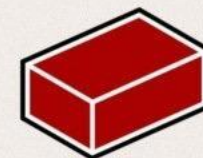
Weechat

- Alternative irc client
- Some advantages over irssi
- Easier to install scripts
- Need to compile it from source, there's a script for that
~edu/compileweechat .
- We need to set our nick for the server
nano ~/.weechat/irc.conf
change "nicks" field to your username under the redbrick section
- Useful commands!
/script
iset.pl - nice interface for changing settings
buffers.py - channel list
screen_away.pl - sets away status when you detach from screen
cmdind.pl -tells you is its command or message
grep - lets you grep in weechat
pushover.pl - notification center
- Remember to */save* when you update settings
- And run */autojoin --run*



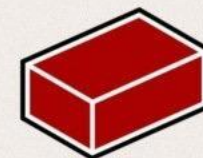
Weechat mobile

- <https://play.google.com/store/apps/details?id=com.ubergeek42.WeechatAndroid&hl=en>
- `/relay add weechat <port number>`
- `/set relay.network.password "your-secret-password"`
- now on the Phone launch the app and in settings In connection settings
 - Relay host: localhost
 - Relay Port: <port number>
 - Relay password: your-secret-password
 - Connect type: SSH tunnel
- then in SSH Tunnel settings
 - SSH host: redbrick.dcu.ie
 - SSH username: your username
 - SSH port: 22



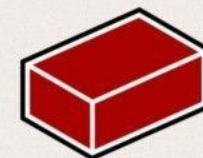
Webspaces

- Open up the FTP Client (Winscp and filezilla)
- Login to
 - Host: `sftp://redbrick.dcu.ie`
 - User: `RedbrickUsername`
 - Pass: `RedbrickPassword`
 - Port: `22`
- Navigate to `public_html/`
- Drag and drop your file to upload or select from the filesystem
- SSH into the Redbrick
- In your home directory (~) enter the following commands:
 - `chmod 711 public_html`
 - `chmod 644 public_html/index.html`



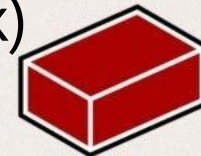
SSH-Keys

- Login without your password
 - On the local machine with linux or osx
 - `ssh-keygen -t rsa`
 - Give it a passphrase
 - `scp ~/.ssh/id_rsa.pub username@login.redbrick.dcu.ie::ssh/`
- Then login in to redbrick
 - `cd .ssh`
 - `touch authorized_keys`
 - `cat id_rsa.pub >> authorized_keys`
 - `rm id_rsa.pub`



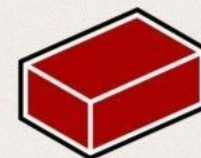
SSH-Keys

- Windows
 - Create a key using PuTTYgen
 - Make a profile on PuTTY with the server settings
 - Note: This will only be saved on your home machine or laptop, won't work on the lab machines unless you download a portable version of PuTTY or PuTTYtray to use
 - Go to Connection -> Data
 - Specify auto-login name
 - We need to generate an SSH-2 RSA key pair with 1024 bits in PuTTYgen
 - Enter whatever as the key comment, and choose a key passphrase
 - Click save public key and choose a location (you could save this on your student drive for the lab machines)
 - Click save private key and do the same (must be .ppk)



SSH-Keys

- Save the public key on the server in a file at `~/.ssh/authorized_keys2`
- Go back to putty and add the private key in SSH -> Auth to the profile you made
 - Remember to save this again
- When you log in you'll need to supply your passphrase.
- You can use Pageant to remember your passphrase for the key
- Pageant will forget the passphrase once you close it, though
- That's that.



Questions?

