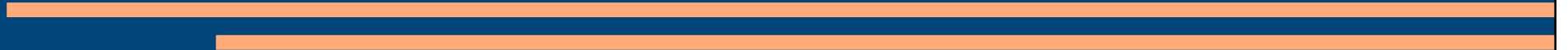


Hands-On UNIX I

... not so gentle introduction



Conventions

type this as root on the command line
(don't type the #)

\$ type this as an ordinary user (don't type the \$)



FreeBSD's installed. What next

- First time you login on your system you are presented with a prompt that looks something like:

```
FreeBSD/i386 (Amnesiac) (ttyv0)
```

```
login:
```

- Enter username 'root' and the password we gave you in class for the root account.
 - CTRL-ALT-FN for virtual terminals, FN is anything from F1 to F8. By default you are in ttyv0 which corresponds to the CTRL-ALT-F1 keyboard sequence.
-
-

Virtual Terminals ctd

- Login and press:
CTRL-ALT-F2
and log in again.
- Try typing in the command
clear
and pressing ENTER. Your screen clears and goes to the top.
Now, you can easily recover your last command by
pressing the UP-ARROW

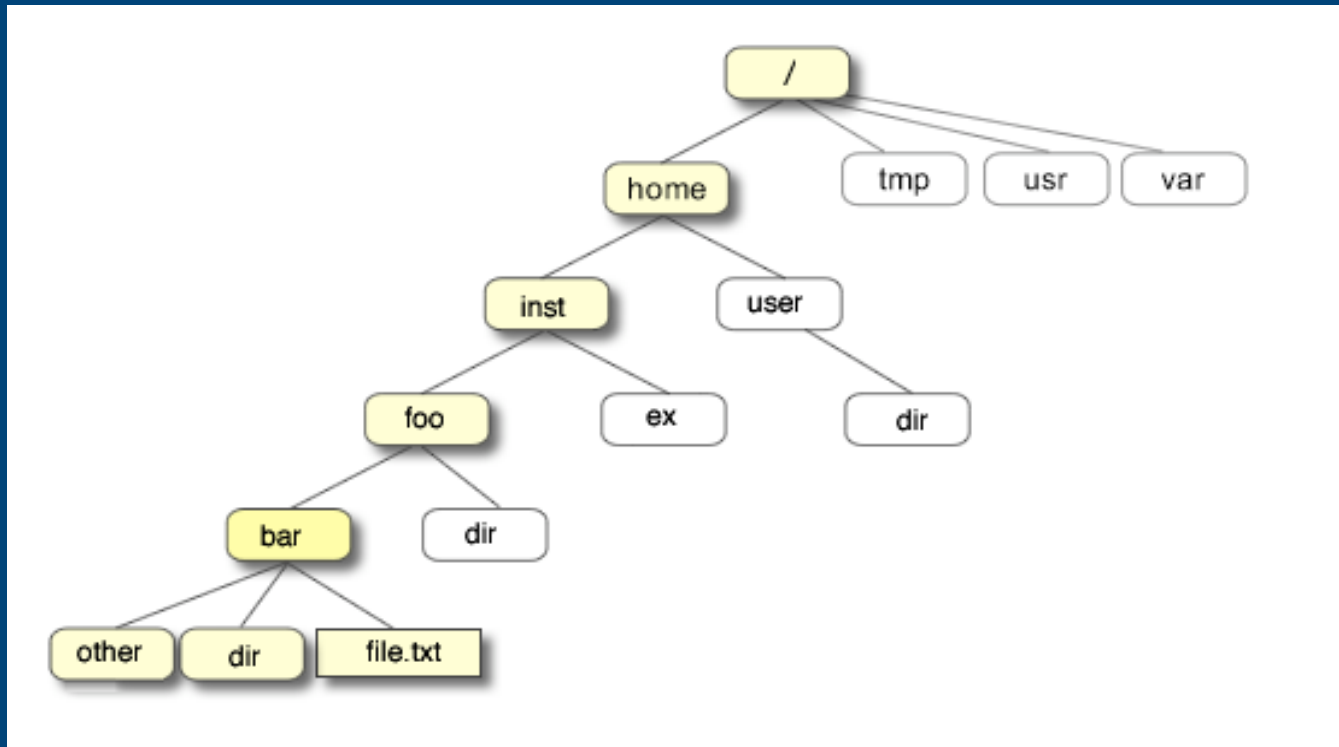
To r00t or not to r00t

- Review UNIX security model
 - uid
 - gid
 - 0 or anything else bypasses all security checks (well except some)
- When to use the root account, and when not to.

Exercise 1: Users



FileSystem: Tree Structure



Exercise 2: Managing Files



Exercise 3: Getting Help



Permissions Revisited

- Three sets of permissions, user, group and world
- Three main permissions: read (r), write (w), execute (x)
- Octal values for each:
 - r = 4
 - w = 2
 - x = 1
- `ls -l` will show the permissions, `chmod` will set them

Using chmod

- Octal set, specify the sum of octal permissions you want to turn on for all permission sets e.g `rwxr-x--` becomes $(4+2+1)(4+1)(0) = 750$... so the command becomes `chmod 750 filename`
- Other way is to specify what set you want and what permission you want to add or remove: `u=user`, `g=group`, `o=other`, `a= all`.
 - `chmod u+w filename` (adds `w` permission to user set on filename – leaves other bits untouched)
 - `chmod u=rx filename` (explicitly sets user bits to `r` and `x` irrespective of what they were)
 - `chmod u+w, g=rx filename` (same thing for more than one set of permissions)

Exercise 4: Editing files with vi



Why vi?

- We teach vi because it's the most common and will be found on almost all UNIX systems. It also happens to be pretty powerful. Other editors are available, but may have to be installed as an additional step: i.e won't come installed by default. A good alternative is ee.

Exercise 5: Editing Files with ee

