

Introduction – Java & JWSDP

- ★ Sun's Java Home Page:
 - ❖ <http://java.sun.com>
- ★ Sun's Developer Network Home Page is
 - ❖ <http://developers.sun.com>
 - ❖ Suggestion:
 - ⊗ Create a login for yourself
 - ⊗ Use it

Introduction – J2SDK and JWSDP

- ★ Java 2 SDK Version 1.4.2 installed in
 - ❖ Windows: c:\j2sdk1.4.2 (.exe)
 - ❖ Linux: /usr/java/j2sdk1.4.2 (.rpm)
- ★ Java 2 SDK documentation installed in
 - ❖ Windows: c:\j2sdk1.4.2\docs (.zip)
 - ❖ Linux: /usr/java/j2sdk1.4.2/docs (.zip)
- ★ Java Web Services Development Pack 1.2
 - ❖ Windows: c:\jwsdp-1.2 (.exe)
 - ❖ Linux: /usr/jwsdp-1.2 (.sh)

Windows Env Vars: J2SDK 1.4.2

- ★ `JAVA_HOME=c:\j2sdk1.4.2`
- ★ `CLASSPATH`: Make sure it contains "." and "`c:\j2sdk1.4.2\jre\classes`"
- ★ `PATH`: Add "`c:\j2sdk1.4.2\bin`"

Windows Env Vars: JWSDP 1.2

- ★ CATALINA_HOME=c:\jwsdp-1.2
- ★ JWSDP_HOME=c:\jwsdp-1.2
- ★ java.endorsed.dirs=c:\jwsdp-1.2\jaxp\lib\endorsed
- ★ PATH: Add "c:\jwsdp-1.2\bin"

Linux Env Vars: J2SDK 1.4.2

- ★ `export JRE_HOME=./usr/java/j2sdk1.4.2/jre`
- ★ `export JAVA_BINDIR=/usr/java/j2sdk1.4.2/jre/bin`
- ★ `export JAVA_HOME=/usr/java/j2sdk1.4.2/jre`
- ★ `export JAVA_ROOT=/usr/java/j2sdk1.4.2`
- ★ `export PATH=$PATH:$JAVA_ROOT/bin`

Linux Env Vars: JWSDP 1.2

- ★ `CATALINA_HOME=/usr/jwsdp-1.2`
- ★ `JWSDP_HOME=/usr/jwsdp-1.2`
- ★ `java.endorsed.dirs=/usr/jwsdp-1.2/jaxp/lib/endorsed`
- ★ `export PATH=$PATH:$JWSDP_HOME/bin`
- ★ File to put commands in:
 - ❖ For All Users on System: `/etc/profile.local`
 - ❖ For Single User Session: `~.bashrc`

Documentation Tool

- ★ Tool **javadoc** documents source code
- ★ Documentation placed in source as prologues
- ★ Can be added easily to Java documentation tree

```
/**  
** This Java program ...  
**  
**/
```

Commands

- ★ Compile: **javac CSServer.java**
- ★ Execute: **java CSServer**
- ★ Document:

```
javadoc .\CSServer.java          ^  
-d .\docs -link c:/j2sdk1.4.2/docs/api -author
```

Note: ^ is a carat and is a continuation character

Development Tools - Batch Files

- ★ Batch Files make a person's life much easier
- ★ **c.bat**: Compile the program
- ★ **go.bat**: Execute the program
- ★ **makedoc.bat**: Make the documentation
- ★ To execute:
 - ❖ Batch_File_Name <CR>
 - ❖ e.g. c:\CSServer> go <CR>

Development Suggestion

- ★ Create folder with same name as program
 - ❖ Place source code and .bats in this folder
 - ❖ Create **.docs** folder under this folder
 - ⊗ **makedoc.bat** creates documentation here
- ★ For each program you write, create a new set up like the one above
- ★ Consider putting all these folders under a folder called **javaprograms**

Example

- ★ Create folder **CSServer** containing
 - ❖ CSServer.java
 - ❖ c.bat, go.bat, makedoc.bat
- ★ Create folder **CSServer\docs** containing
 - ❖ Documentation resulting from makedoc.bat
 - ❖ Top level file is **index.html**

CSServer.java Example

- ★ Creates a simple window with
 - ❖ a message area on the bottom
 - ❖ a text input area on the top
- ★ Waits for a client to call at its socket port
- ★ Can do 3 things
 - ❖ Echo messages to output received from client
 - ❖ Send messages to client
 - ❖ End the session and wait for another client when a client sends TERMINATE (all caps)

CSCClient.java Example

- ★ Requests a sockets connection to the server
- ★ Creates a simple window like the server
- ★ Can do 2 things
 - ❖ Echo messages to output received from the server
 - ❖ Send a message to the server
- ★ Run Example Programs

Manual Protocols

- ★ using PuTTY try **www.stny.rr.com**, try **loopback too**
- ★ Found port: 7, echo
- ★ Found port: 9, discard
- ★ Found port: 13, daytime
- ★ Found port: 17, qotd
- ★ Found port: 19, chargen
- ★ Found port: 21, ftp
- ★ Found port: 25, smtp
- ★ Found port: 110, pop3

Manual SMTP Example

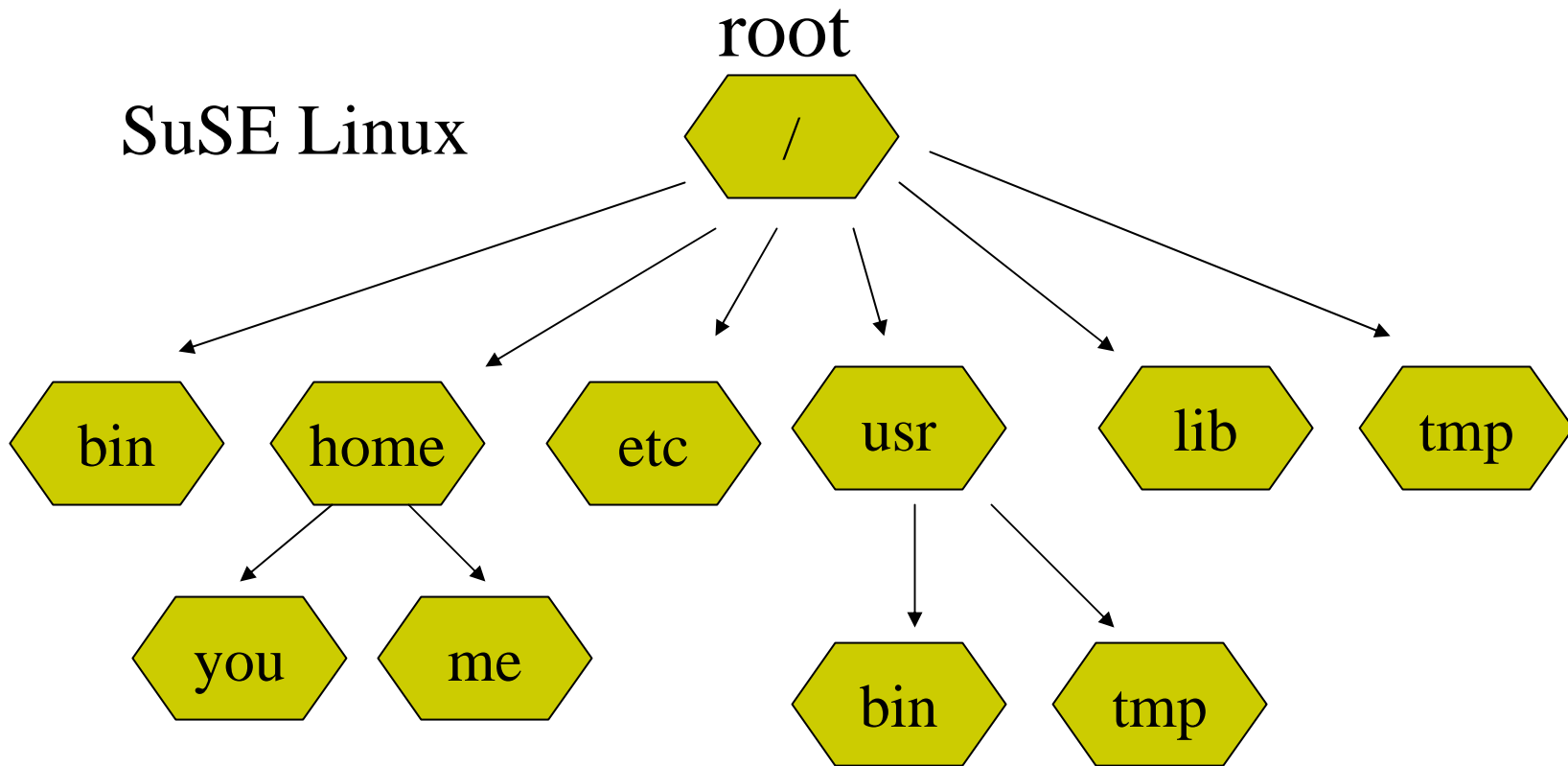
- ★ C:Successful connection to Host: "smtp-server.stny.rr.com/ 24.92.226.153", Port: "25" with a TCP socket connection
- ★ S:220 ms-smtp-03.nyroc.rr.com **AUTHORIZED USE ONLY**
- ★ C:helo stny.rr.com
- ★ S:250 ms-smtp-03.nyroc.rr.com Hello bgm-24-95-152-200. stny.rr.com [24.95.152.200]
- ★ C:mail from:<ryder@stny.rr.com>
- ★ S:250 2.1.0 <ryder@stny.rr.com>... Sender ok
- ★ C:rcpt to:<ryderj@oneonta.edu>
- ★ S:250 2.1.5 <ryderj@oneonta.edu>... Recipient ok
- ★ C:data
- ★ S:354 Please start mail input.
- ★ C:Here is line 1
- ★ C:This is line 2 - Single dot on next line
- ★ C:.
- ★ S:250 Mail queued for delivery.
- ★ C:quit

Linux - History

★ Some History

- ❖ Unix originated at AT&T in late 1960's
- ❖ "Linux is an operating system that was initially created as a hobby by a young student, **Linus Torvalds**, at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in **1991** when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The current full-featured version is 2.4 (released January 2001) and development continues."
(From <http://www.linux.org/info>)

Tree Directory Structure



Mouse and KDE GUI World

- ★ The MCS Lab has SuSE Linux 8.2
- ★ Linux mouse has 3 buttons (not 2)
- ★ Fileviewer
- ★ Konqueror
- ★ Mounting devices (Diskette, CD, Zip, etc.)
 - ❖ `mount /media/cdrom`
- ★ Games
- ★ Terminal Sessions

Bourne Shell Commands

- ★ `passwd` Change password
- ★ `ls [-al]` List contents of directory
- ★ `pwd` Print working directory
- ★ `cat` Type contents of file to screen
- ★ `mkdir/rmdir` Create/Remove directory
- ★ `cd` Change directory
- ★ `touch` Create a file
- ★ `cp, mv, rm` Copy, Rename, Remove file
- ★ `chmod` Change file permissions
- ★ `emacs` EMACS file editor

.bashrc

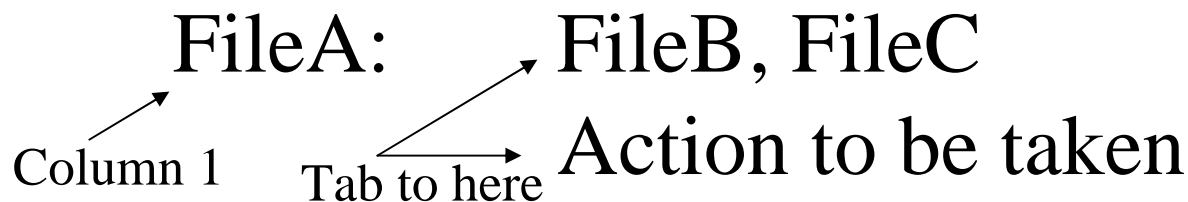
- ★ Executes every time you create a new terminal session
- ★ Way to tailor your personal environment the way you like it
- ★ `export PS1=Bob`
- ★ `export PATH=.:$PATH`

Java Example

- ★ **env** shows environment variables
- ★ Program location:
 - ❖ `~/javaprograms/SimpleApplication`
- ★ Compile: `javac SimpleApplication.java`
- ★ Make Java documentation:
 - ❖ `javadoc ./SimpleApplication.java` ^
 `-d ./docs -link /usr/java/j2sdk1.4.2/docs/api -author`
- ★ Execute: `java SimpleApplication`

Make Utility

- ★ Way to simplify development
- ★ If date on FileB or FileC is newer than date on FileA, they actions will be performed
- ★ FileA must begin in column 1
- ★ You must tab over then type FileB and others
- ★ '\' allows you to continue onto the next line



Make Utility

- ★ File name must be **makefile** in current directory
- ★ Run the makefile with the command **make**
- ★ Each comparison will be made and if true the actions will be performed
- ★ Once dependencies set up properly, only those files needing recompile will be recompiled