

Lesson Module Status

- Slides – draft
- Properties - done
- Flash cards – na
- First minute quiz – done
- Web calendar summary – done
- Web book pages – done
- Commands – done
- Lab – done
- Supplies () - na
- Class PC's – na
- Chocolates - bringing

- CCC Confer wall paper – done
- labx1 and project posted - done

- Materials uploaded – done
- Backup headset charged – nope
- Backup slides, CCC info, handouts on flash drive - done

- Check that room headset is charged – done



Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**



Emanuel



Tanner



Merrick



Quinton



Chris



Bobby



Craig



Jeff



Yu-Chen



Terry



Tommy



Eric



Dan M



Geoffrey



Marisol



Josh



Gabriel



Jesse



Tajvia



Daniel W



Jason

The LAST Quiz

Please close your books, notes, lesson materials, forum and answer these questions **in the order** shown:

- What must you do to a variable so a child can use it?
- How would you use an alias to make a command named copy ... that would do what the cp command does?
- How do you make an alias setting permanent?

email answers to: risimms@cabrillo.edu



- [] Has the phone bridge been added?
- [] Is recording on?
- [] Does the phone bridge have the mike?
- [] Share slides, putty (rsimms, simmsben, simmsben), and Chrome, and VirtualBox Eko VM
- [] Disable spelling on PowerPoint

Printing

Objectives	Agenda
<ul style="list-style-type: none">• Be able to print, view the print queue and cancel print jobs	<ul style="list-style-type: none">• Quiz• Housekeeping• Refresh• Printing



Questions

Previous material and assignment

1. Previous material
2. Lab 10



Housekeeping



Previous material and assignment

1. Lab 10 due midnight tonight
2. Grades Page – please check progress and grade choice
3. Extra Credit Labs X1 and X2 (30 points each)
4. Calendar endgame
5. Forum code tagging

Fall 2010 Linux Classes

Select	Term	Status	Section Name and Title	Location	Meeting Information	Faculty	Available/ Capacity/ Waitlist	Credits	CEUs
<input type="checkbox"/>	Fall 2011	Open	CIS-191AB-73605 (73605) UNIX/Linux Inst. Config. Admin	Main Campus	08/29/2011-12/17/2011 Lecture Tuesday 06:00PM - 08:05PM, Computer Information Labs, Room 2501 (more)...	J. Griffin	12 / 24 / 0	4.00	
<input type="checkbox"/>	Fall 2011	Open	CIS-192A-73604 (73604) UNIX/Linux TCP/IP Admin	Main Campus	10/25/2011-12/13/2011 Lecture Tuesday 01:00PM - 05:10PM, Computer Information Labs, Room 2501 (more)...	R. Simms	18 / 24 / 0	2.00	

1) The above two classes will be offered on the same day, but the 192A is only for the second 8 weeks

2) CIS 191 is a hybrid class that will meet 2 hours a week (Tuesday evening) in the classroom and another two hour session from a lecture archive which the student can choose when to view. Labs are also part of the class and that time can be done remotely and scheduled to the student's convenience.

If there are a few students who could not possibly make the CIS 191 two hours on campus, they can contact Jim for possible ways to make the class still work.



Stuff

Silence is golden

Many UNIX commands that run successfully produce no output

```
[roddyduk@opus bin]$ alias details=file  
[roddyduk@opus bin]$ cp quiet quiet.bak  
[roddyduk@opus bin]$ umask 002  
[roddyduk@opus bin]$ cat quiet > /dev/null  
[roddyduk@opus bin]$ > important_file
```

Silence is golden

Running or sourcing a script full of UNIX commands that produce no output still produces no output!

```
[roddyduk@opus bin]$ cat quiet  
alias details=file  
cp quiet quiet.bak  
umask 002  
cat quiet > /dev/null
```

```
[roddyduk@opus bin]$ quiet  
[roddyduk@opus bin]$ source quiet  
[roddyduk@opus bin]$
```

Silence is golden

Shell script developers will use the echo command to provide some indication of status or progress with the scripts they write.

```
[roddyduk@opus bin]$ cat quiet
alias details=file
cp quiet quiet.bak
umask 002
cat quiet > /dev/null
echo "Quiet script successfully completed"
```

```
[roddyduk@opus bin]$ quiet
Quiet script successfully completed
[roddyduk@opus bin]$ source quiet
Quiet script successfully completed
[roddyduk@opus bin]$
```



final project permissions

Final Project

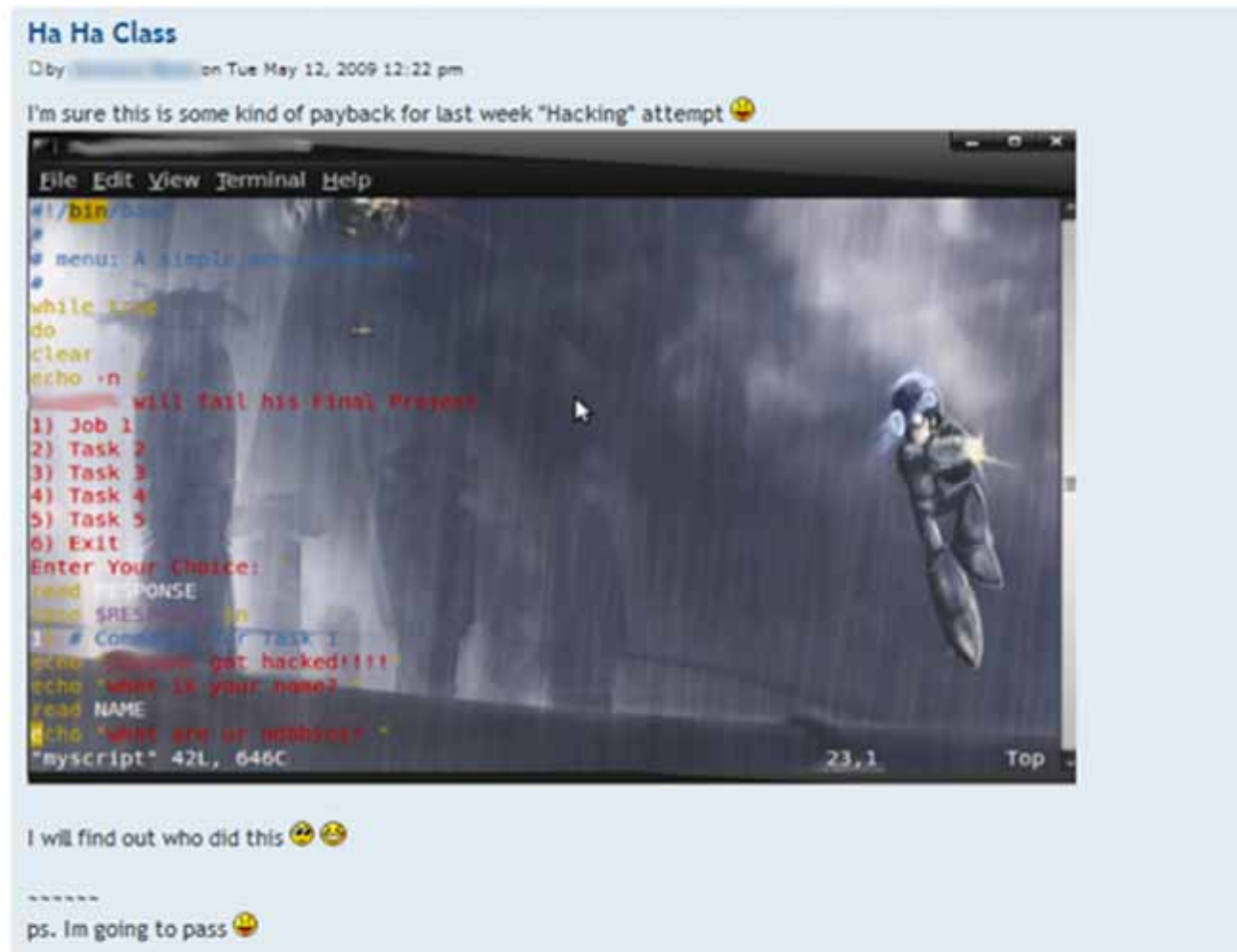
```
simmsben@opus:~  
/home/cis90ol/simmsben $ allscripts  
  
*****  
*           Spring 2011 CIS 90 Online Projects           *  
*****  
1) Bobby      7) Eric      13) Josh      19) Terry  
2) Chris      8) Gabriel    14) Marisol   20) Tommy  
3) Craig      9) Geoffrey   15) Merrick  21) Yu-Chen  
4) Dan M.     10) Jason     16) Quinton  
5) Daniel W.  11) Jeff      17) Tajvia  
6) Emanuel    12) Jesse     18) Tanner  
  
*****  
*           Examples and Hall of Fame           *  
*****  
50) Duke      51) Benji     52) Junious   53) Janet  
  
99) Exit  
  
Enter Your Choice: █
```

*Before leaving class today you want to make sure you can run your script from **allscripts***

Benji went a tad overboard with some of his scripts (he will do anything for some chicken)

Permissions

A past forum post ...



Uh, oh ... someone got hacked!

Permissions

```

simmsben@opus:~
/home/cis90ol/simmsben $ ls -l /home/cis90ol/*/bin/myscript
-rwxr-xr-x 1 christan cis90ol 676 May 5 15:10 /home/cis90ol/christan/bin/myscript
-rwxr-xr-x 1 dienequi cis90ol 518 May 5 18:18 /home/cis90ol/dienequi/bin/myscript
-rwxrwxr-x 1 elmenchr cis90ol 663 May 5 15:10 /home/cis90ol/elmanchr/bin/myscript
-rwxrwxr-x 1 hextcra cis90ol 645 May 5 15:11 /home/cis90ol/hextcra/bin/myscript
-rwxrwxr-x 1 hillejef cis90ol 631 May 5 15:10 /home/cis90ol/hillejef/bin/myscript
-rwxrwxr-x 1 keezeter cis90ol 482 May 5 15:15 /home/cis90ol/keezeter/bin/myscript
-rwxrwxr-x 1 lighttom cis90ol 729 May 5 15:10 /home/cis90ol/lighttom/bin/myscript
-rwxrwxr-x 1 mcnamdan cis90ol 672 May 5 15:10 /home/cis90ol/mcnamdan/bin/myscript
-rwxrwxrwx 1 paytomar cis90ol 628 May 5 15:10 /home/cis90ol/paytomar/bin/myscript
-rwxr-x--- 1 roddyduk cis90ol 3198 May 2 09:24 /home/cis90ol/roddyduk/bin/myscript
-rwxr-x--x 1 simmsben cis90ol 10490 May 5 07:52 /home/cis90ol/simmsben/bin/myscript
-rwxrwxr-x 1 sylvijos cis90ol 635 May 5 15:10 /home/cis90ol/sylvijos/bin/myscript
-rw-rw-r-- 1 warrejes cis90ol 481 May 5 14:55 /home/cis90ol/warrejes/bin/myscript
-rwxrwxr-x 1 willitaj cis90ol 671 May 5 15:10 /home/cis90ol/willitaj/bin/myscript
-rwxr-xr-x 1 wilsodan cis90ol 640 May 5 15:10 /home/cis90ol/wilsodan/bin/myscript
-rwxrwxr-x 1 wingejas cis90ol 770 May 5 15:11 /home/cis90ol/wingejas/bin/myscript
/home/cis90ol/simmsben $ █

```

Which myscrip files can only be edited by their owner? Which ones could be edited by anyone in the CIS 90 class? Which ones could be edited by anyone on Opus?

Note: One of the requirements for the final project is setting permissions on your script so that all cis900l members can run it.

For example, you could meet this requirement by going to your bin/ directory and issuing:

chmod 750 myscript

It's up to you if you also want to give the group write access or provide others on Opus with any access at all.

Permissions

Why can other classmates write to my scripts?

Before Lab 10

```
/home/cis90/roddyduk/bin $ umask
0002
/home/cis90/roddyduk/bin $ rm newscript; touch newscript
/home/cis90/roddyduk/bin $ ls -l newscript
-rw-rw-r-- 1 roddyduk cis90 0 Nov 23 16:17 newscript
/home/cis90/roddyduk/bin $ chmod +x newscript
/home/cis90/roddyduk/bin $ ls -l newscript
-rwxrwxr-x 1 roddyduk cis90 0 Nov 23 16:17 newscript
```

After Lab 10

```
/home/cis90ol/simmsben $ umask
0006
/home/cis90ol/simmsben $ rm newscript; touch newscript
/home/cis90ol/simmsben $ ls -l newscript
-rw-rw---- 1 simmsben cis90ol 0 May 12 08:44 newscript
/home/cis90ol/simmsben $ chmod +x newscript
/home/cis90ol/simmsben $ ls -l newscript
-rwxrwx--x 1 simmsben cis90ol 0 May 12 08:44 newscript
```

Because your umask setting gives group member write permission on any new files you create!

Permissions

```
[roddyduk@opus bin]$ cat /home/cis90/roddyduk/.bash_profile
```

```
# .bash_profile
```

```
# Get the aliases and functions
```

```
if [ -f ~/.bashrc ]; then
```

```
    . ~/.bashrc
```

```
fi
```

```
# User specific environment and startup programs
```

```
PATH=$PATH:$HOME/../../bin:$HOME/bin:.
```

```
BASH_ENV=$HOME/.bashrc
```

```
USERNAME=""
```

```
PS1='$PWD $ '
```

```
export USERNAME BASH_ENV PATH
```

```
umask 002
```

```
set -o ignoreeof
```

```
stty susp
```

```
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `
```

Note your umask is defined in .bash_profile which runs every time you login. In lab 10 your change this setting to 006.



- Change your umask to 026
- Can group or other users modify your new files now?
- Try it, **touch** a new file and check the permissions with **ls -l**
- How would you make this a permanent umask setting?



dates

Fun with Dates

```
/home/cis90/roddyduk $ date  
Wed Nov 26 15:35:53 PST 2008
```

```
/home/cis90/roddyduk $ date +%m/%d/%y'  
11/26/08
```

```
/home/cis90/roddyduk $ date +%m/%d/%Y'  
11/26/2008
```

```
/home/cis90/roddyduk $ date +%m/%d/%Y and %N nanoseconds'  
11/26/2008 and 334957229 nanoseconds
```

```
/home/cis90/roddyduk $ date +'Time: %H hours and %M minutes'  
Time: 15 hours and 41 minutes
```

```
/home/cis90/roddyduk $ man date
```

See the man page for lots of other % sequences

Write a short script, named mydate, that prints out a greeting followed by the date in the mm/dd/yyyy format

Hint: Use vi to make a file that includes one of the commands on the previous slide. Save this file and give it execute permission. Test it!

The output could look like this:

```
/home/cis90ol/simmsben $ mydate  
Hello simmsben  
05/12/2011  
/home/cis90ol/simmsben $
```



tips on script names

Don't name your scripts "script"

```
[roddyduk@opus bin]$ ls -l script  
-rwxr-x--- 1 roddyduk cis90 47 Nov 23 16:44 script
```

```
[roddyduk@opus bin]$ cat script  
echo "Hello from the script file named script"
```

What would happen if you ran the script above?

Don't name your scripts "script"

```
[roddyduk@opus bin]$ cat script
echo "Hello from the script file named script"
```



```
[roddyduk@opus bin]$ script
Script started, file is typescript
```



*Why the heck
doesn't my script
do what it's
supposed to do?*

```
[roddyduk@opus bin]$ Where is my script?
bash: Where: command not found
[roddyduk@opus bin]$ exit
Script done, file is typescript
[roddyduk@opus bin]$ cat typescript
Script started on Wed 13 May 2009 08:00:02 AM PDT
[roddyduk@opus bin]$ Where is my script?
bash: Where: command not found
[roddyduk@opus bin]$ exit
```

```
Script done on Wed 13 May 2009 08:00:47 AM PDT
[roddyduk@opus bin]$
```

Don't name your scripts "script"

Why doesn't script do what it is supposed to do? ... because script is the name of an existing UNIX command!

```
[roddyduk@opus bin]$ man script
[roddyduk@opus bin]$
```

The screenshot shows a terminal window titled "roddyduk@opus:~/bin" with a window manager title bar. The terminal displays the output of the command "man script". The output is formatted as follows:

```
SCRIPT (1) BSD General Commands Manual SCRIPT (1)
NAME
    script - make typescript of terminal session
SYNOPSIS
    script [-a] [-c COMMAND] [-f] [-q] [-t] [file]
DESCRIPTION
    Script makes a typescript of everything printed on your terminal. It is
    useful for students who need a hardcopy record of an interactive session
    as proof of an assignment, as the typescript file can be printed out
    later with lpr(1).

    If the argument file is given, script saves all dialogue in file. If no
    file name is given, the typescript is saved in the file typescript.

Options:
    -a      Append the output to file or typescript, retaining the prior con-
           tents.
    -c COMMAND
           Run the COMMAND rather than an interactive shell. This makes it
           easy for a script to capture the output of a program that behaves
           differently when its stdout is not a tty.
```

Don't name your scripts "script"

There are (at least) two files named script on Opus

```
[roddyduk@opus bin]$ type script
script is hashed (/usr/bin/script)
[roddyduk@opus bin]$ file /usr/bin/script
/usr/bin/script: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs),
for GNU/Linux 2.6.9, stripped
```

```
[roddyduk@opus bin]$ type /home/cis90/roddyduk/bin/script
/home/cis90/roddyduk/bin/script is /home/cis90/roddyduk/bin/script
[roddyduk@opus bin]$ file /home/cis90/roddyduk/bin/script
/home/cis90/roddyduk/bin/script: ASCII text
[roddyduk@opus bin]$
```

Question: *Why did bash run the script in /usr/bin instead of the script in /home/cis90/roddyduk/bin?*

Don't name your scripts "script"

Question: Why did bash run the script in /usr/bin instead of the script in /home/cis90/roddyduk/bin?

*The Linux **script** command is in this directory*

```
[roddyduk@opus bin]$ echo $PATH  
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/bin:  
/home/cis90/roddyduk/bin:.
```

*Our script, named **script**, is in this directory*

Answer: bash searches the path in the order the directories are listed. It finds the script command in /user/bin first.

Don't name your scripts "script"

To override the PATH you can always specify an absolute pathname to the file you want to run:

```
[roddyduk@opus bin]$ /home/cis90/roddyduk/bin/script  
Hello from the script file named script
```

```
[roddyduk@opus bin]$ ./script  
Hello from the script file named script
```

Note the shell treats the . above as "here" which in this case is /home/cis90/roddyduk/bin



Try the script command

- Use the **script** command to start recording
- Type various commands of your choice
- Type **exit** or hit Ctrl-D to end recording
- Use **cat typescript** to see what you recorded

This would be a good way to record a session such as working one of the lab assignments for future reference.



Refresh



Process Life Cycle

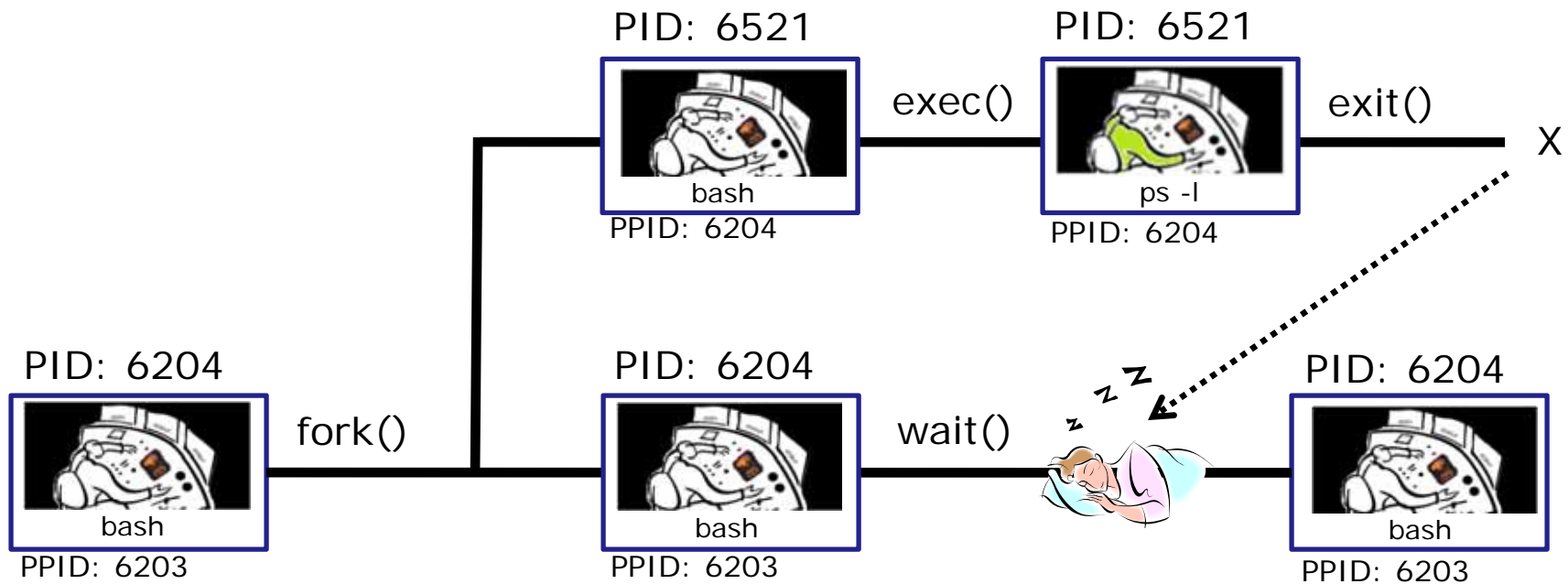
An example process

Running the `ps` command

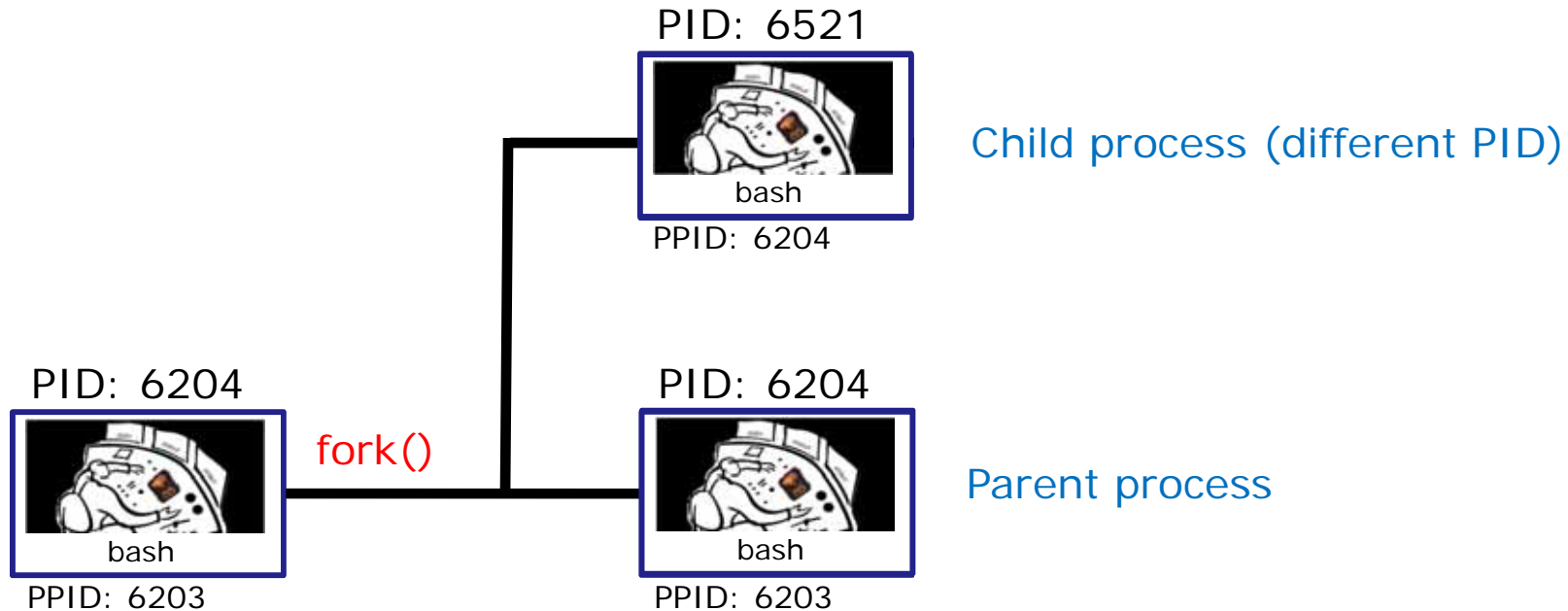
```
[rsimms@opus ~]$ ps -l
F S  UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  201  6204  6203  0  75   0  -   1165  wait  pts/6      00:00:00 bash
0 R  201  6521  6204  0  77   0  -   1050  -     pts/6      00:00:00 ps
```

*Whenever you run any command, program, or script it runs as a **child process***

Example Process Lifecycle



Example Process Lifecycle

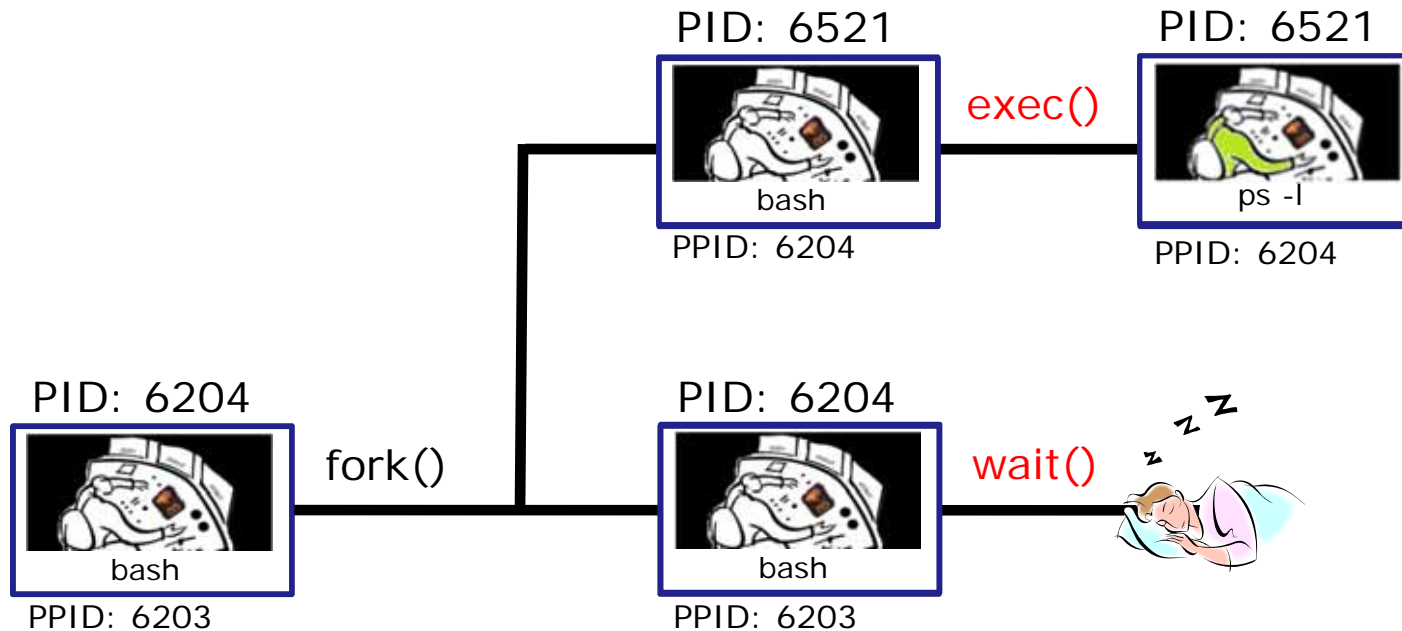


1) When a program is loaded into memory a new process must be created.

This is done by the **parent** process (in this case **bash**) making a copy of itself using the fork system call.

The new **child** process is a duplicate of the **parent** but it has a different PID.

Example Process Lifecycle



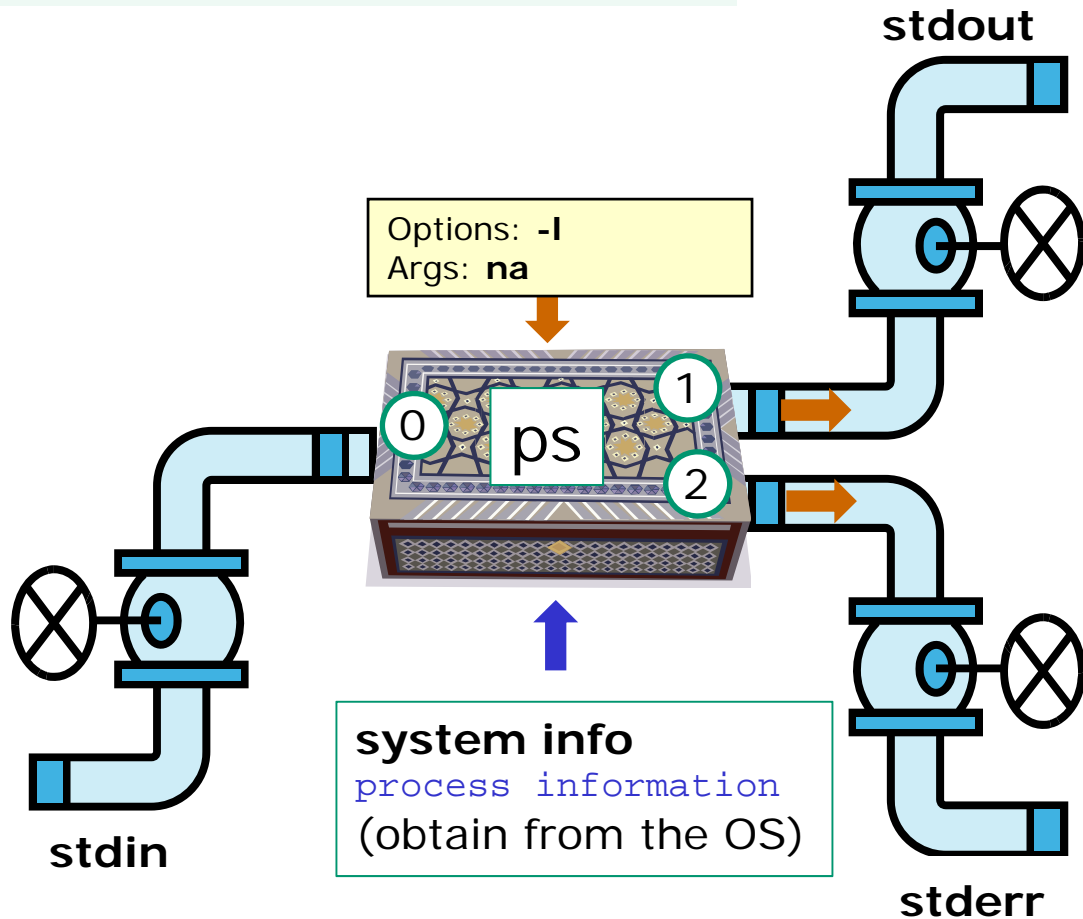
2) An `exec` system call is issued to overlay the **child** process with the instructions of the requested command (in this case the **ps** command). The new instructions then are executed.

The **parent** process issues the `wait` system call and goes to sleep.

The **ps** command instructions have been exec'ed and are now running in memory as a process connected via the file descriptors stdin, stdout and stderr

```
$ ps -l
```

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 201 6204 6203 0 75 0 - 1165 wait pts/6 00:00:00 bash
0 R 201 6521 6204 0 77 0 - 1050 - pts/6 00:00:00 ps
```

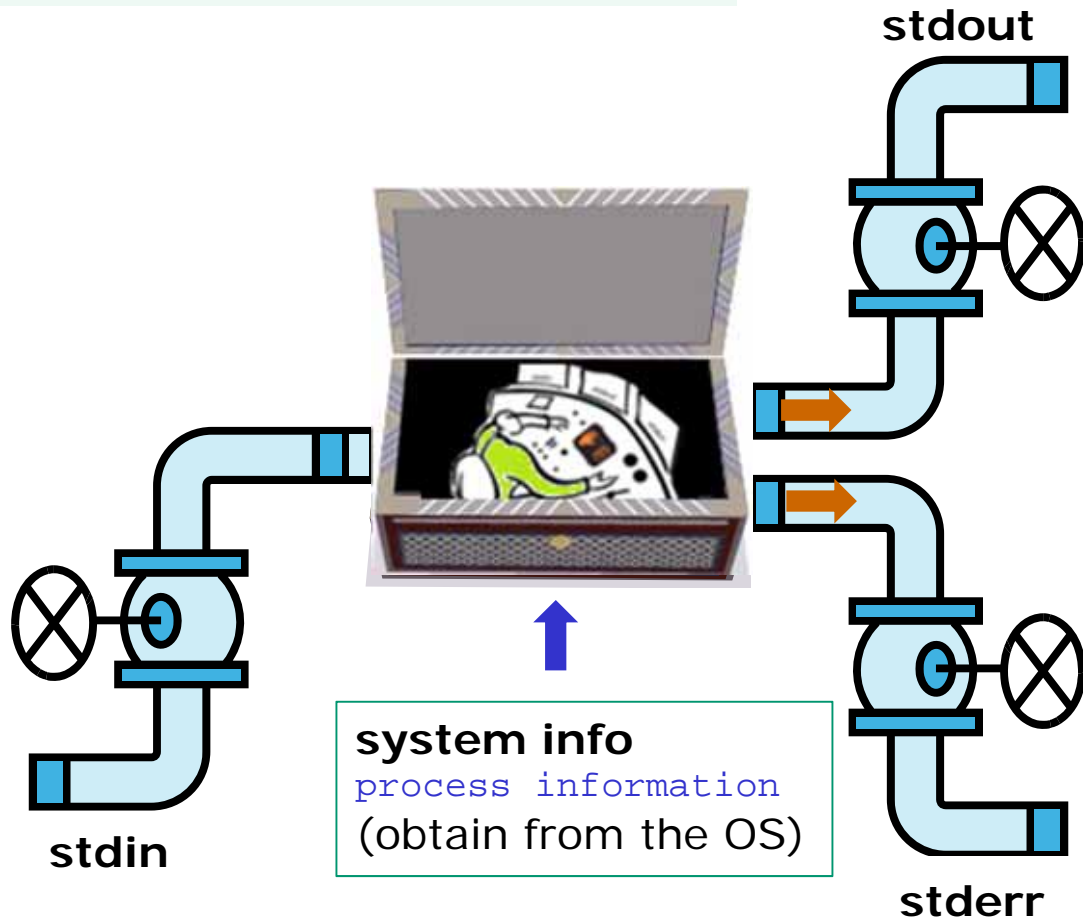


*In this example, output from the **ps** command goes to stdout.*

stdout has not been redirected so it goes to the default terminal device (your screen).

\$ ps -l

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	201	6204	6203	0	75	0	-	1165	wait	pts/6	00:00:00	bash
0	R	201	6521	6204	0	77	0	-	1050	-	pts/6	00:00:00	ps

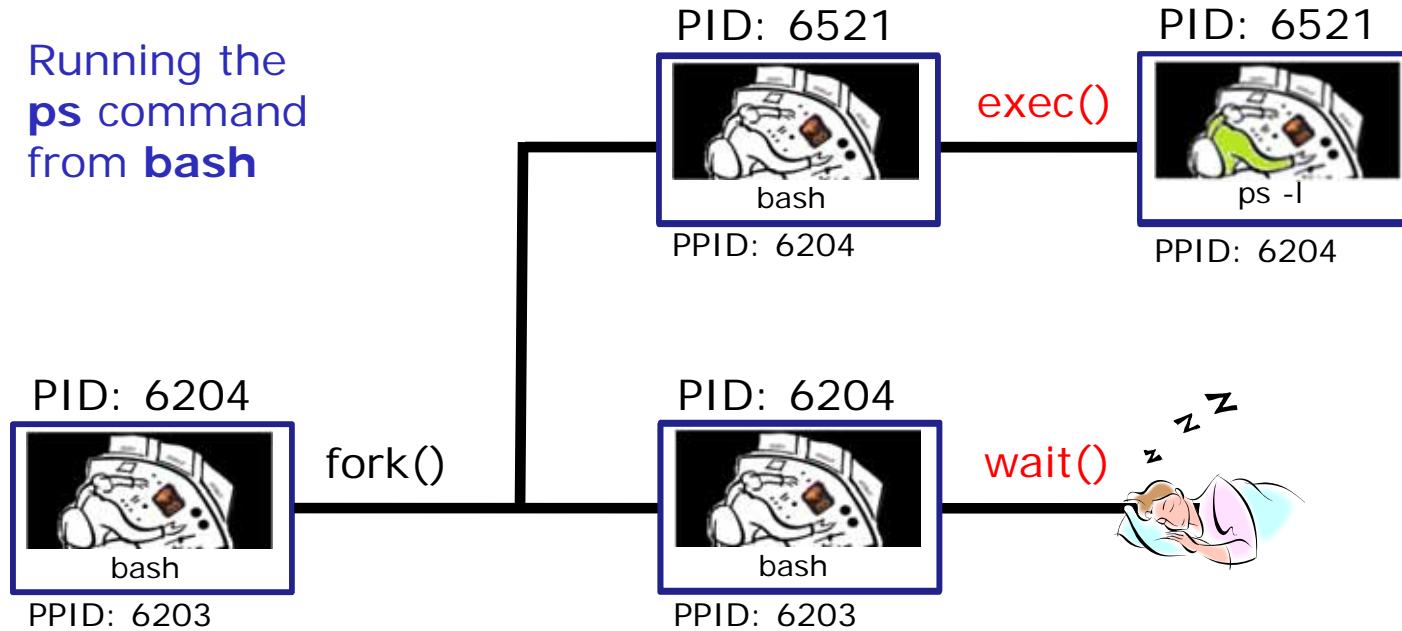


A sneak peek into memory to see what our process looks like!



Process Lifecycle

Running the **ps** command from **bash**



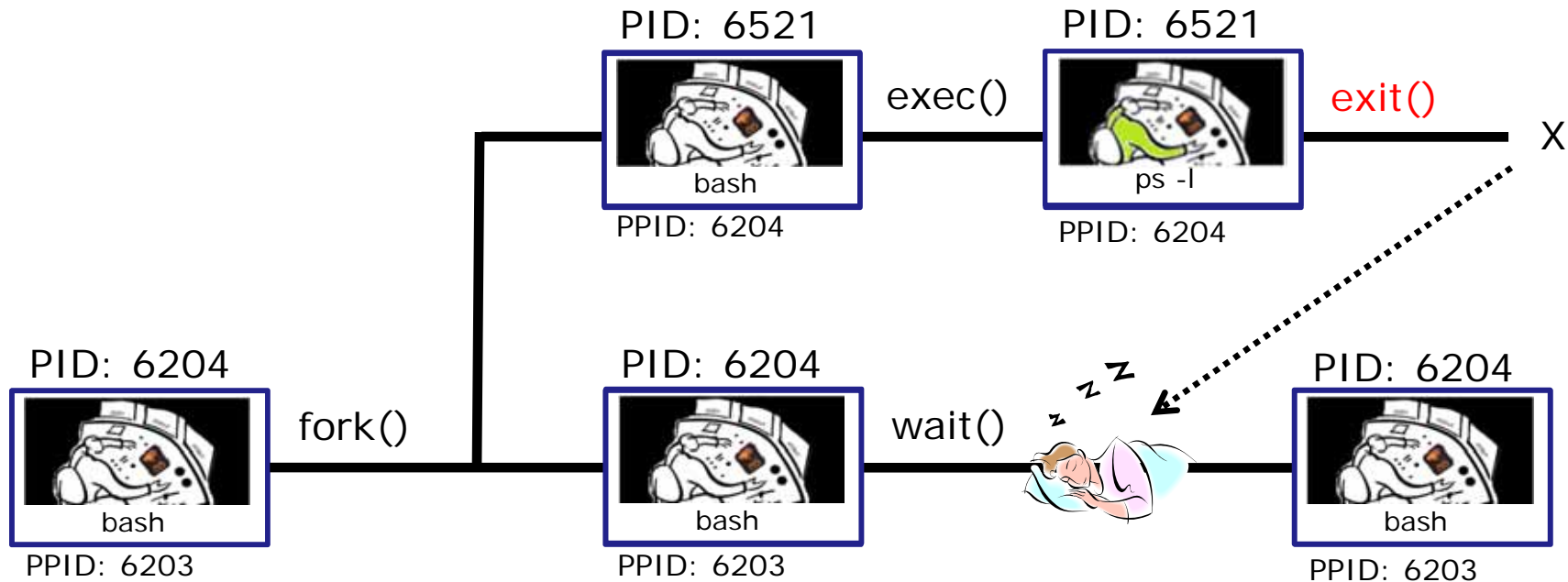
```
[rsimms@opus ~]$ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	201	6204	6203	0	75	0	-	1165	wait	pts/6	00:00:00	bash
0	R	201	6521	6204	0	77	0	-	1050	-	pts/6	00:00:00	ps

2) An **exec** system call is issued to overlay the **child** process with the instructions of the requested command (in this case the **ps** command). The new instructions then are executed.

The **parent** (in this case **bash**) process issues the **wait** system call and goes to sleep.

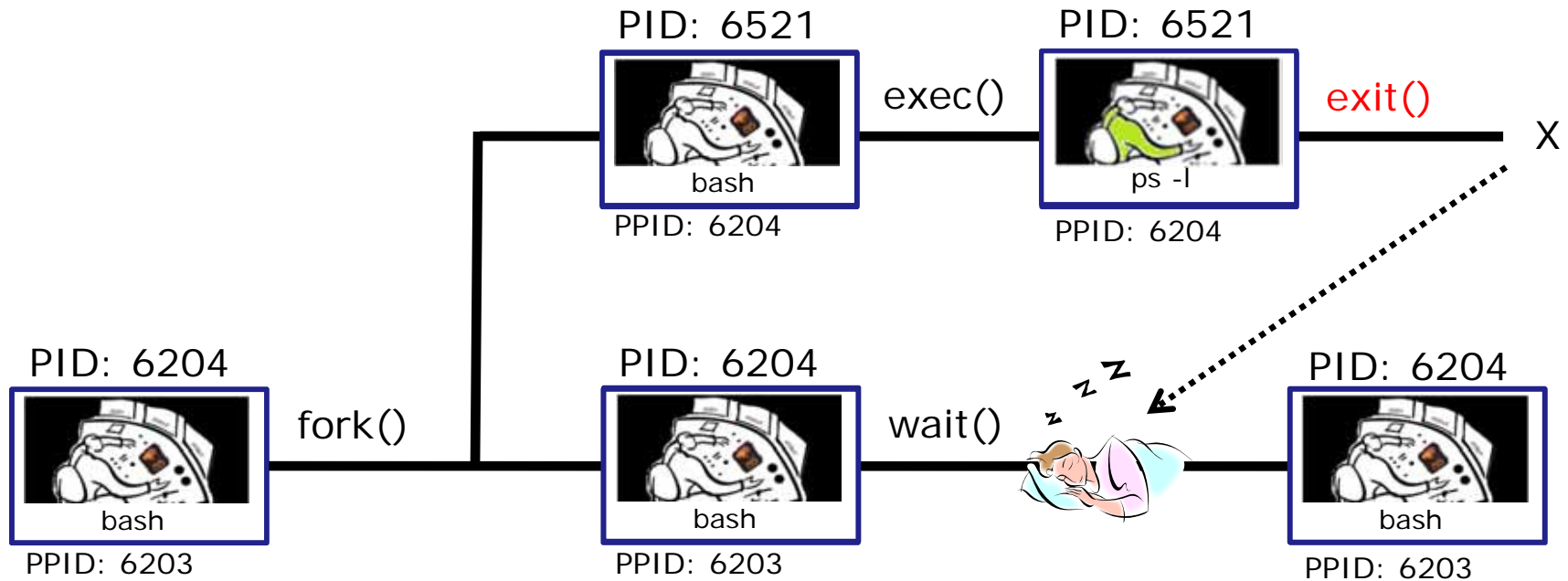
Process Lifecycle



3) When the **child** process (in this case **ps** command) finishes executing the instructions it issues the **exit** system call. At this point it gives up all its resources becomes a **zombie**.

The **parent** (in this case **bash**) is woken up and once the **parent** has informed the kernel it has finished working with the **child**, the **child** process is killed and removed from the process table.

Process Lifecycle



3) If the **parent** process were to die before the **child**, the zombie will become an **orphan**. Fortunately the init process will adopt any orphaned **zombies**.



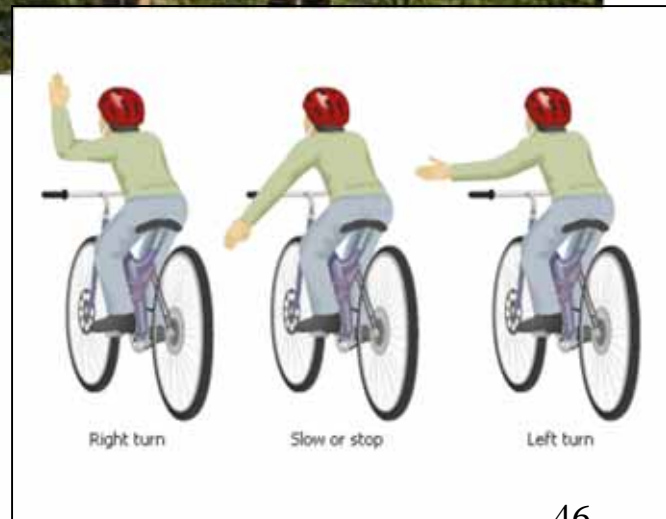
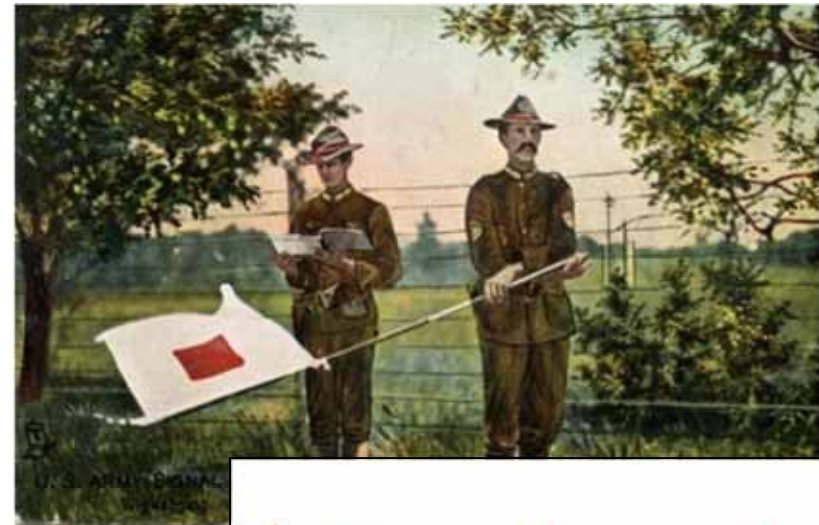
Signals

Signals

PLATE 4

COMMERCIAL CODE SIGNALS		
EXAMPLES OF THE SEVERAL HOISTS WHICH CAN BE MADE HAVING TWO, THREE, OR FOUR FLAGS.		
When a word contains two letters of the same name, the second time of its occurrence it must begin or be in the 2nd hoist; and on the 3rd occurrence, it must begin or be in the 3rd hoist.		
URGENT & IMPORTANT SIGNALS	COMPASS SIGNALS	3 FLAGS
CODE FLAG OVER 1 FLAG OR 2 FLAG SIGNALS		
CODE FLAG P	A	A
"I Am about to Sail"	C	K
"Do Not"		X
"Abandon the Vessel"	N/E	S/SW
LATITUDE & LONGITUDE SIGNALS	CODE FLAG OVER 2 FLAGS	
CODE FLAG A	Q	Q
OR O	H	E
12° Latitude	X	OR H
North Latitude		Y
		Z
		23° Longitude
		East Longitude
NUMERAL TABLE	GENERAL VOCABULARY	GEOGRAPHICAL SIGNALS ALPHABETICAL ORDER
CODE FLAG UNDER 2 FLAGS	3 FLAG SIGNAL	4 FLAG SIGNAL
Y	I	A
S	X	E
CODE FLAG	K	Y
10,000		Z
		Glasgow, Scotland.
ALPHABETICAL SPELLING TABLE	NAMES OF VESSELS FROM CODE LIST	
J	C	H
O	B	C
H	D	L
N	N	B
John	Abb	oil
		Glasgow of Glasgow
		1058 Tons # 32036

JAMES BROWN & SON GLASGOW



Signals

Signals are asynchronous messages sent to processes

They can result in one of three courses of action:

1. be ignored,
2. default action (die)
3. execute some predefined function.

Signals are sent:

- Using the kill command: **\$ kill -# PID**
 - Where # is the signal number and PID is the process id.
 - if no number is specified, SIGTERM is sent.
- Using special keystrokes
 - limited to just a few signals

Use kill -l to see all signals

Signals

Signals are asynchronous messages sent to processes



Asynchronous means it can happen at any time

Signals

SIGHUP	1	Hangup (POSIX)
SIGINT	2	Terminal interrupt (ANSI) Ctrl-C
SIGQUIT	3	Terminal quit (POSIX) Ctrl-\
SIGILL	4	Illegal instruction (ANSI)
SIGTRAP	5	Trace trap (POSIX)
SIGIOT	6	IOT Trap (4.2 BSD)
SIGBUS	7	BUS error (4.2 BSD)
SIGFPE	8	Floating point exception (ANSI)
SIGKILL	9	Kill (POSIX) can't be caught or ignored
SIGUSR1	10	User defined signal 1 (POSIX)
SIGSEGV	11	Invalid memory segment access (ANSI)
SIGUSR2	12	User defined signal 2 (POSIX)
SIGPIPE	13	Write on a pipe with no reader, Broken pipe (POSIX)
SIGALRM	14	Alarm clock (POSIX)
SIGTERM	15	Termination (ANSI) default kill signal when not specified

Use kill -l to see all signals

Signals

SIGSTKFLT	16	Stack fault
SIGCHLD	17	Child process has stopped or exited, changed (POSIX)
SIGCONT	18	Continue executing, if stopped (POSIX)
SIGSTOP	19	Stop executing (can't be caught or ignored) (POSIX)
SIGTSTP	20	Terminal stop signal (POSIX) <i>Ctrl-Z or Ctrl-F</i>
SIGTTIN	21	Background process trying to read, from TTY (POSIX)
SIGTTOU	22	Background process trying to write, to TTY (POSIX)
SIGURG	23	Urgent condition on socket (4.2 BSD)
SIGXCPU	24	CPU limit exceeded (4.2 BSD)
SIGXFSZ	25	File size limit exceeded (4.2 BSD)
SIGVTALRM	26	Virtual alarm clock (4.2 BSD)
SIGPROF	27	Profiling alarm clock (4.2 BSD)
SIGWINCH	28	Window size change (4.3 BSD, Sun)
SIGIO	29	I/O now possible (4.2 BSD)
SIGPWR	30	Power failure restart (System V)

Use kill -l to see all signals



Try and kill one of your login sessions

- Start up a second session on Opus
- Use `ps -u $LOGNAME`
- Kill the second session from the first session
- Use `kill <-#> <PID>`
- Which process did you target? (bash, sshd, ...)
- Which signal did you send? (default, -9, ...)



Aliases

alias command

(a shell builtin)

```
alias [-p] [name[=value] ...]
```

Alias with no arguments or with the `-p` option prints the list of aliases in the form `alias name=value` on standard output. When arguments are supplied, an alias is defined for each name whose value is given. A trailing space in value causes the next word to be checked for alias substitution when the alias is expanded. For each name in the argument list for which no value is supplied, the name and value of the alias is printed. Alias returns true unless a name is given for which no alias has been defined.

Note aliases are not expanded by default in non-interactive shell, and it can be enabled by setting the `expand_aliases` shell option using `shopt`.

alias command

showing all aliases

```
/home/cis90/roddyduk $ alias
alias bill='cd /home/cis90/roddyduk/poems/Shakespeare'
alias bye='clear;exit'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias me='finger roddyduk'
alias print='echo -e'
alias rm='rm -i'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
/home/cis90/roddyduk $
```

*Typing **alias** by itself will show all your current aliases*

alias command

creating a new alias

```
/home/cis90/roddyduk $ alias s="clear; head -10 ~/edits/small_town"  
/home/cis90/roddyduk $ s  
HOW SMALL IS SMALL?
```

YOU KNOW WHEN YOU'RE IN A SMALL TOWN WHEN...

The airport runway is terraced.

The polka is more popular than a mashpit on on Saturday night.

Third Street is on the edge of town.

Every sport is played on dirt.

The editor and publisher of the newspaper carries a camera at all times.

You don't use your turn signal because everyone knows where you are
going knows where you are going.

*Make an alias, called **s**, that prints the first 10 lines of smalltown*

alias command

showing and deleting an alias

```
/home/cis90/roddyduk $ alias s="clear; head -10 ~/edits/small_town"
```

Using the alias command to make an alias

```
/home/cis90/roddyduk $ type s  
s is aliased to `clear; head -10 ~/edits/small_town'  
/home/cis90/roddyduk $ alias s  
alias s='clear; head -10 ~/edits/small_town'
```

Using the type or alias command to show an existing alias

```
/home/cis90/roddyduk $ unalias s  
/home/cis90/roddyduk $
```

Using unalias command to remove an alias

alias command

strong (') or weak (") quote marks

\$ ac=on
\$ fan=medium
\$ defrost=off

double →

```

$ alias p="echo $ac $fan $defrost"
$ alias p
alias p='echo on medium off'
$ p
on medium off
$ ac=off
$ p
on medium off
        
```

← *single*

```

$ alias p='echo $ac $fan $defrost'
$ alias p
alias p='echo $ac $fan $defrost'
$ p
on medium off
$ ac=off
$ p
off medium off
        
```

Very subtle: using strong quotes (') prevents bash from expanding the variables when setting up the alias



Make this alias which we will use later:

```
alias show='echo fan=$fan ac=$ac; type copy; env | grep ac'
```



Shell Variables

Shell Variables

- Shell variables are names consisting of alpha-numeric characters.
- Variables defined by the Operating System are uppercase, e.g. TERM, PS1, PATH
- The **set** command will display the shell's current variables and their values.
- Shell variables are initialized using the assignment operator:
TERM=vt100
Note: Quotes must be used for white space: **VALUE="any value"**
- Variables may be viewed using the echo command: **echo \$TERM**
The \$ in front of a variable name denotes the value of that variable.
- To remove the value from a variable, use the unset command:
unset PS1
- Shell variables hold their values for the duration of the session i.e. until the shell is exited



Environment Variables

Environment Variables

- A subset of the shell variables are environment variables.
- Environment variables are shell variables that have been exported.
- The **env** command will display the current environment variables and their values. Using the **export** command by itself will also show all the environment variables.
- The **export** command is used to make a shell variable into an environment variable. E.g. **dog=benji; export dog** creates a new environment variable named dog.
- The **export -n** command is used to make an environment variable back into a normal shell variable. E.g. **export -n dog** makes dog back into a regular shell variable.
- Child processes are provided copies of the parent's environment variables. Any changes made by the child will not effect the parent's copies.

Common Environment Variables

Shell Variable	Description
HOME	Users home directory (starts here after logging in and returns with a <code>cd</code> command (with no arguments))
LOGNAME	User's username for logging in with.
PATH	List of directories, separated by ':'s, for the Shell to search for commands (which are program files) .
PS1	The prompt string.
PWD	Current working directory
SHELL	Name of the Shell program being used.
TERM	Type of terminal device , e.g. dumb, vt100, xterm, ansi, etc.

On Opus, PS1 is set in /etc/bashrc and then redefined in .bash_profile

Environment Variables

env command – show all environment variables

```
[roddyduk@opus ~]$ env
HOSTNAME=opus.cabrillo.edu
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
SSH_CLIENT=63.249.103.107 20807 22
SSH_TTY=/dev/pts/0
USER=roddyduk
LS_COLORS=no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05
;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=
00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00
;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=
00;35:*.png=00;35:*.tif=00;35:
USERNAME=
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/roddyduk/./bin:/home/cis90/roddyduk/bin:
.
MAIL=/var/spool/mail/roddyduk
PWD=/home/cis90/roddyduk
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
fan=medium
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HOME=/home/cis90/roddyduk
SHLVL=2
BASH_ENV=/home/cis90/roddyduk/.bashrc
LOGNAME=roddyduk
CVS_RSH=ssh
SSH_CONNECTION=63.249.103.107 20807 207.62.186.9 22
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/bin/env
[roddyduk@opus ~]$
```

These are all shell variables that have been exported and they are available to child processes

Environment Variables

export command – show all exported variables

```
[rododyduk@opus ~]$ export
```

```
declare -x BASH_ENV="/home/cis90/rododyduk/.bashrc"
```

```
declare -x CVS_RSH="ssh"
```

```
declare -x G_BROKEN_FILENAMES="1"
```

```
declare -x HISTSIZE="1000"
```

```
declare -x HOME="/home/cis90/rododyduk"
```

```
declare -x HOSTNAME="opus.cabrillo.edu"
```

```
declare -x INPUTRC="/etc/inputrc"
```

```
declare -x LANG="en_US.UTF-8"
```

```
declare -x LESSOPEN="|/usr/bin/lesspipe.sh %s"
```

```
declare -x LOGNAME="rododyduk"
```

```
declare -x
```

```
LS_COLORS="no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:"
```

```
declare -x MAIL="/var/spool/mail/rododyduk"
```

```
declare -x OLDPWD
```

```
declare -x
```

```
PATH="/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/rododyduk/../../bin:/home/cis90/rododyduk/bin:."
```

```
declare -x PWD="/home/cis90/rododyduk"
```

```
declare -x SHELL="/bin/bash"
```

```
declare -x SHLVL="2"
```

```
declare -x SSH_ASKPASS="/usr/libexec/openssh/gnome-ssh-askpass"
```

```
declare -x SSH_CLIENT="63.249.103.107 20807 22"
```

```
declare -x SSH_CONNECTION="63.249.103.107 20807 207.62.186.9 22"
```

```
declare -x SSH_TTY="/dev/pts/0"
```

```
declare -x TERM="xterm"
```

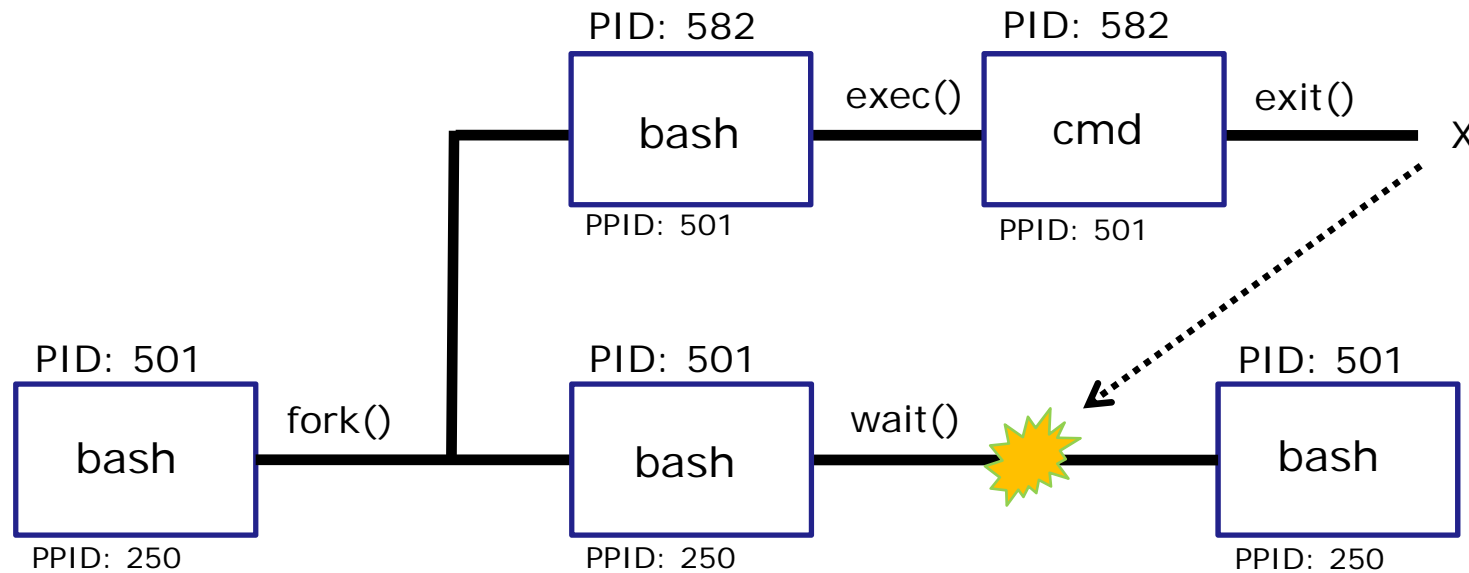
```
declare -x USER="rododyduk"
```

```
declare -x USERNAME=""
```

```
[rododyduk@opus ~]$
```

These are all shell variables that have been exported and they are available to child processes

Children only see exported (environment) variables



When a shell forks a child, not all of the variables get passed on to the child. Only those the environment variables (which have been exported) are passed on to the child.

- Use **env** to see all the environment variables
- Use **export** to make a shell variable an environment variable and available to child processes e.g. **export BIRTHDAY**



Shell Environment

Customizing the shell environment

- It possible to customize your shell environment by editing the hidden **.bash_profile** and **.bashrc** files in your home directory.
- You can create and initialize shell variables.
- You can modify existing environment variables, e.g. PATH and PS1
- You can create new environment variables.
- You can modify or add new aliases
- You can specify the umask setting
- You can run commands or scripts

bash startup files

*only
executed
when
logging in*

/etc/profile (all)

- o adds root's special path

/etc/profile.d/*.sh (all)

- o kerberos directories added to path
- o adds color, vi aliases
- o language, character sets

.bash_profile (user specific)

- o adds user's bin to path

.bashrc (user specific)

- o add aliases here

*To permanently
customize your shell
environment you
modify these home
directory files in Lab 10*

/etc/bashrc (all)

- o changes umask to 0002 for regular users
- o sets final prompt string

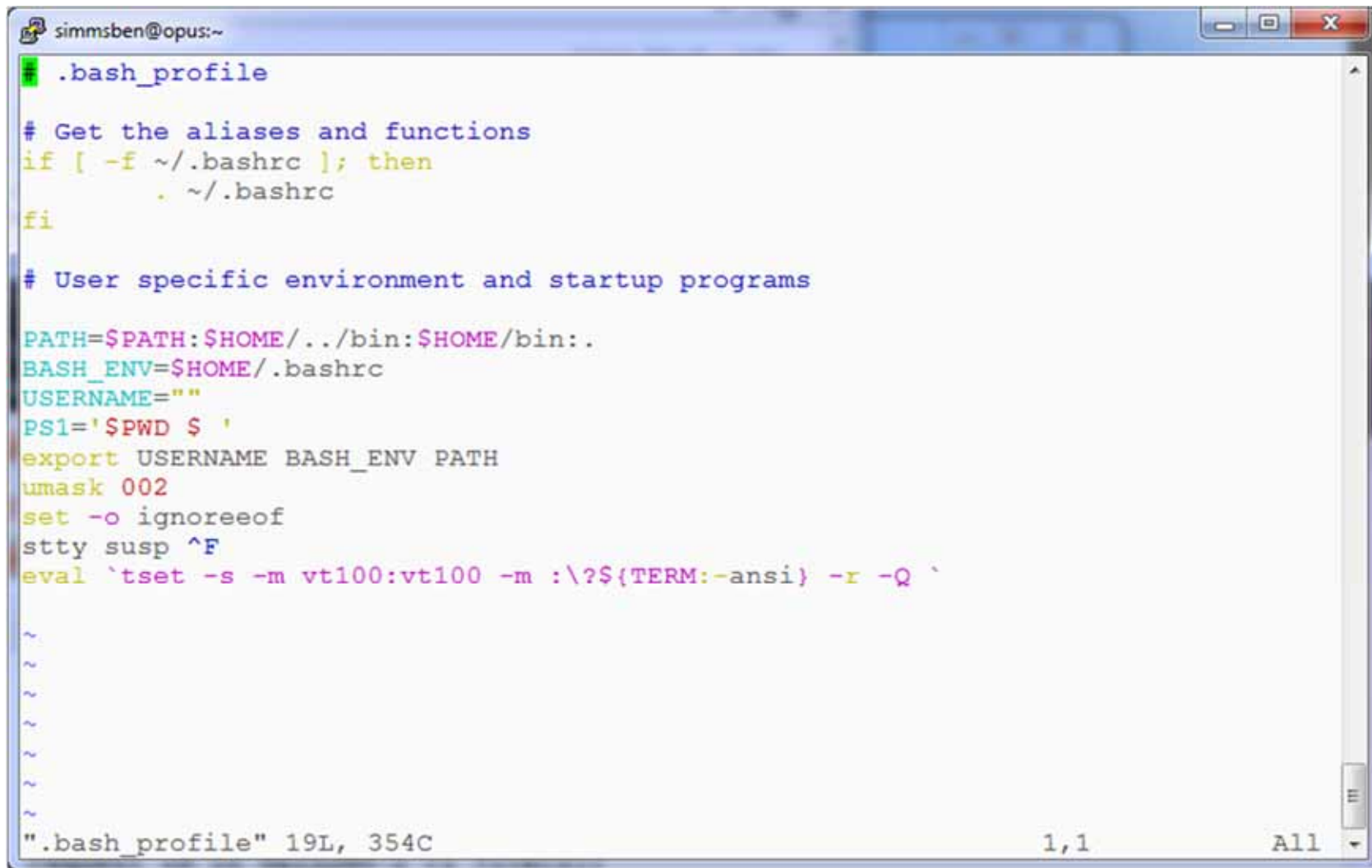


.bash_profile

.bash_profile

- The `.bash_profile` is a shell script that sets up a user's shell environment.
- This script is run (sourced) each time the user logs in.
- The `.bash_profile` is used for initializing shell variables, running the user's `.bashrc` file, running basic commands like `umask` and `set -o` options.
- `.bash_profile` is not run for sub-shells (child processes)

.bash_profile for CIS 90 accounts



```
simmsben@opus:~
. bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/../bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=""
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 002
set -o ignoreeof
stty susp ^F
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `

~
~
~
~
~
~
~

".bash_profile" 19L, 354C                        1,1      All
```

Before doing Lab 10

.bash_profile for CIS 90 accounts

```

rododyduk@opus:~
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:/home/cis90/bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=""
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 006
set -o ignoreeof
stty susp ^F
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `
mesg n
BIRTHDAY=05/05/93
export BIRTHDAY
riddle
~
~
~
~
16,1 All

```

After doing Lab 10

.bashrc

.bashrc

The `.bashrc` is a shell script that is executed during user login and whenever a new shell is invoked.

- This script is run (sourced) each time the user logs in.
- The `.bashrc` is typically used for defining aliases
- `.bashrc` is run for sub-shells (e.g. using the `bash` command to start a new sub-shell)



. and exec

. and exec

In normal execution of a unix command, shell-script or binary, the child process is unable to affect the login shell environment.

Sometimes it is desirable to run a shell script that will initialize or change shell variables in the parent environment. To do this, the shell (bash) provides a `.` (dot) or **source** command, which instructs the shell to execute the shell script itself, without spawning a child process to run the script.

`.` *myscript* or `source` *myscript*

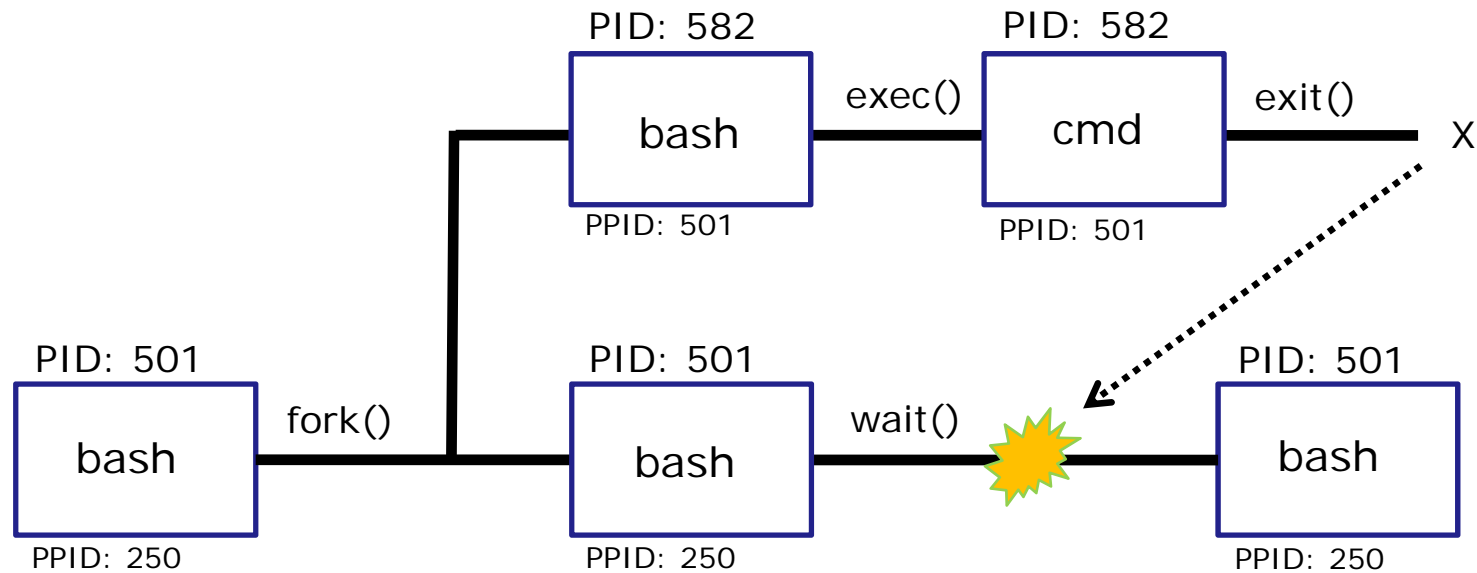
In this example, the commands in the file `shscript` are run by the parent shell, and therefore, any changes made to the environment will last for the duration of the login session.

If a UNIX command is run using the `exec` command, the shell will terminate upon the exiting of that command:

`exec clear`

This will have the effect of clearing the screen and logging off the computer.

Children can not change the parent's variables



When a shell forks a child, not all of the variables get passed on to the child. Only those the environment variables (which have been exported) are passed on to the child.

- The child gets a copy of the parents environment variables
- Changes made to the copies do not change the parent's variables

. and exec

```
/home/cis90/roddyduk $ cat setupvars
```

```
echo "program is being run"
fan=high
ac=on
export ac
alias copy=cp
```

A sample script to create some variables and an alias. Note only one variable is exported.

```
/home/cis90/roddyduk $ echo fan=$fan ac=$ac; type copy; env | grep ac
fan= ac=
-bash: type: copy: not found
```

Initial state

```
/home/cis90/roddyduk $ setupvars
```

```
program is being run
```

```
/home/cis90/roddyduk $ echo fan=$fan ac=$ac; type copy; env | grep ac
fan= ac=
-bash: type: copy: not found
```

Not changed!

```
/home/cis90/roddyduk $ source setupvars
```

```
program is being run
```

```
/home/cis90/roddyduk $ echo fan=$fan ac=$ac; type copy; env | grep ac
fan=high ac=on
copy is aliased to `cp`
ac=on
```

Changed when sourced!

A child cannot make changes to the parent, use source or . when you need a script to make changes.

. and exec

parent

```
/home/cis90/roddyduk $ bash Start a sub-shell  
child process
```

```
[roddyduk@opus ~]$ echo fan=$fan ac=$ac; type copy; env | grep ac  
fan= ac=on  
bash: type: copy: not found  
ac=on
```

Only the exported variables exist for the child

child

```
[roddyduk@opus ~]$ . setupvars  
program is being run  
[roddyduk@opus ~]$ echo fan=$fan ac=$ac; type copy; env | grep ac  
fan=high ac=on  
copy is aliased to `cp`  
ac=on
```

. can be used for the source command

parent

```
[roddyduk@opus ~]$ exec setupvars  
program is being run  
/home/cis90/roddyduk $
```

exec replaces bash code with program script. When finished the child is killed

We are back in the parent shell because we used exec. If we had not been a child process our session would have abruptly ended!

print command (alias)

What is going on ??????????????????

Make a print alias for lp, then try it in a sub-shell (child process) and the behavior completely changes!

```
/home/cis90/roddyduk $ alias print=lp
/home/cis90/roddyduk $ print lab10
request id is hplaser-9 (1 file(s))
/home/cis90/roddyduk $ bash
```

*The **lp** command is used to print files on a printer*

child

```
[roddyduk@opus ~]$ ls lab10
lab10
[roddyduk@opus ~]$ print lab10
lab10
[roddyduk@opus ~]$ print A B C $LOGNAME
A B C roddyduk
```

*Huh? Why is print now behaving as if it were the **echo** command instead of the **lp** command*

What is going on ??????????????????

```
[roddyduk@opus ~]$ type print
print is aliased to `echo -e'
[roddyduk@opus ~]$ alias print
alias print='echo -e'
[roddyduk@opus ~]$ cat .bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
alias print="echo -e"
alias bye="clear;exit"
alias rm="rm -i"
alias bill="cd /home/cis90/$LOGNAME/poems/Shakespeare"
[roddyduk@opus ~]$
```

child

*Our print alias was changed! It is no longer aliased to the **lp** command*

.bashrc is sourced when starting a new sub-shell and this reset the alias!

Moral of the story is ...

child

```
[roddyduk@opus ~]$ exit
exit
/home/cis90/roddyduk $ type print
print is aliased to `lp'
/home/cis90/roddyduk $ print lab10
request id is hplaser-10 (1 file(s))
/home/cis90/roddyduk $
```

When we exit the sub-shell our new print alias is back in action

Moral of the story is, aliases do not get exported like environment variables. If you want an alias to be available in a child process you must add it to .bashrc



Printers

Sneak Peak for CIS 90 Students





- Two predominate types of printers*
- *Thermal inkjet technology*
 - *Laser, drum, toner technology*



So many ways to hook them up ...

Now:

- *Network*
- *USB*
- *Wireless (Bluetooth, IR)*



Back then:

- *Serial cable*
- *Parallel printer cable*



Printer Configuration



Printing Commands

System V based print subsystem

- lp (to print)
- lpstat (queue management)
- cancel (to remove jobs)

BSD based print subsystem

- lpr (to print)
- lpq (queue management)
- lprm (to remove jobs)

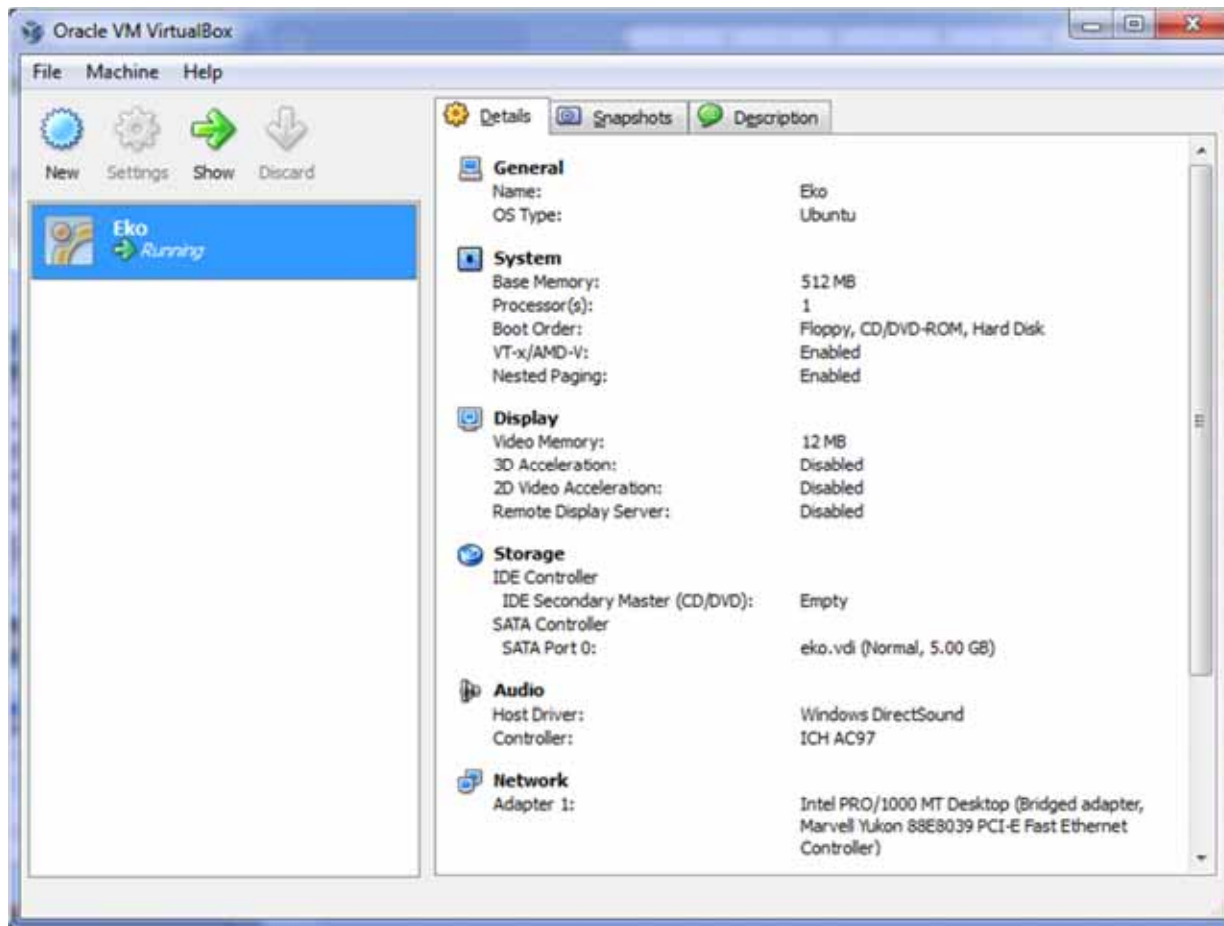
CUPS

- Provides both System V and Berkeley based command-line interfaces
- Supports new Internet Printing Protocol
- Works with Samba

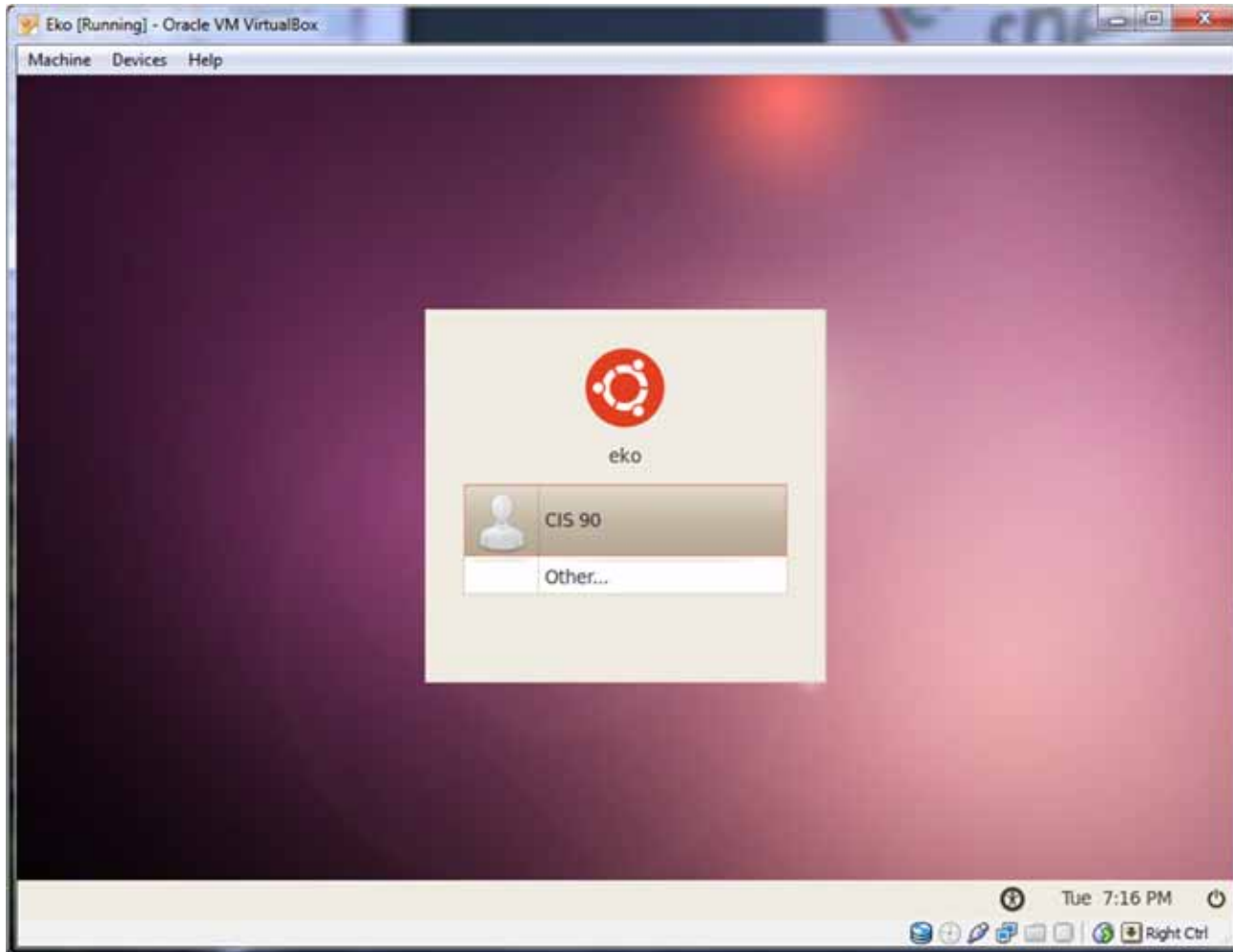
We will be just looking at CUPS

CUPS

For the lesson on printing we will be using the Eko virtual machine.

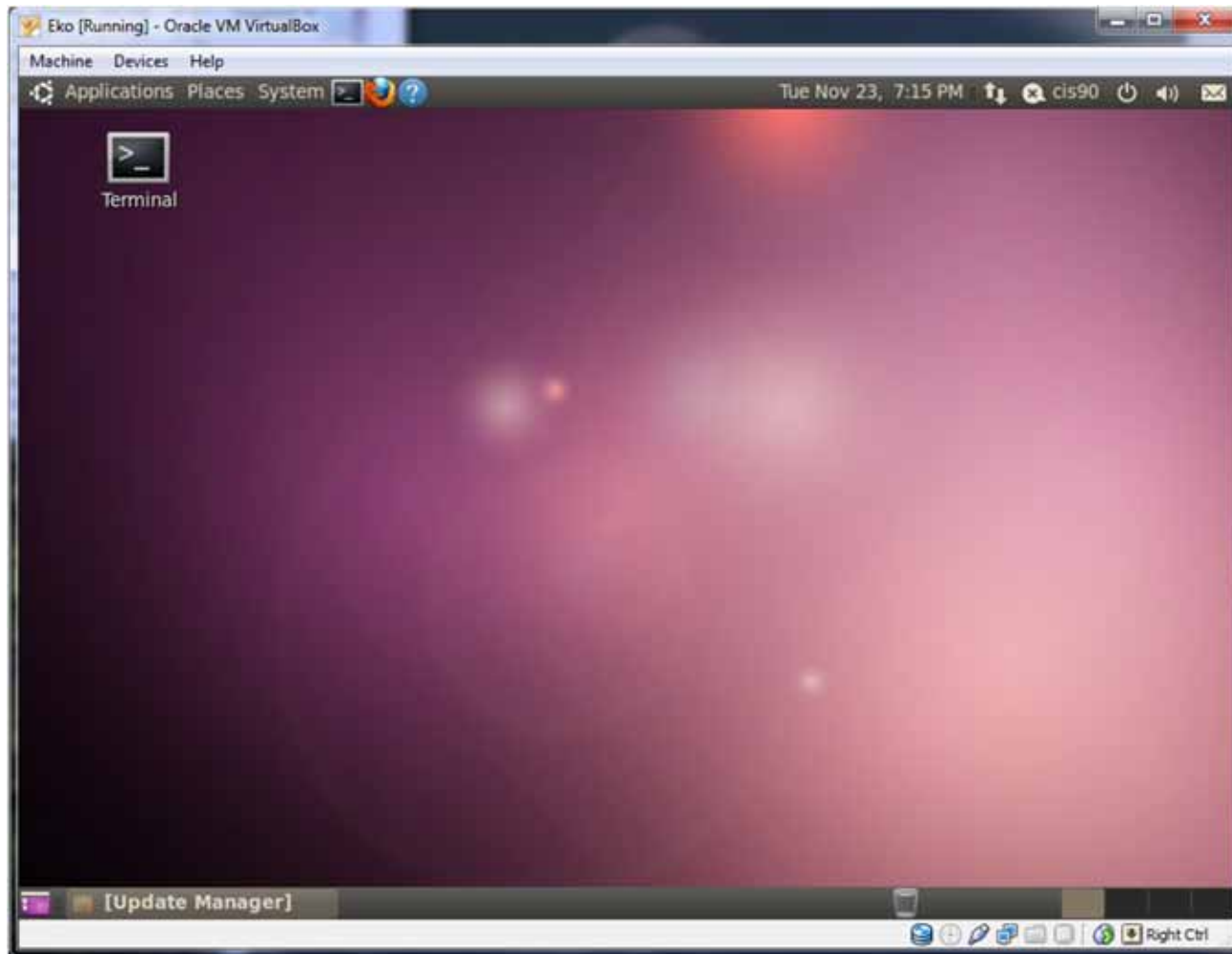


CUPS



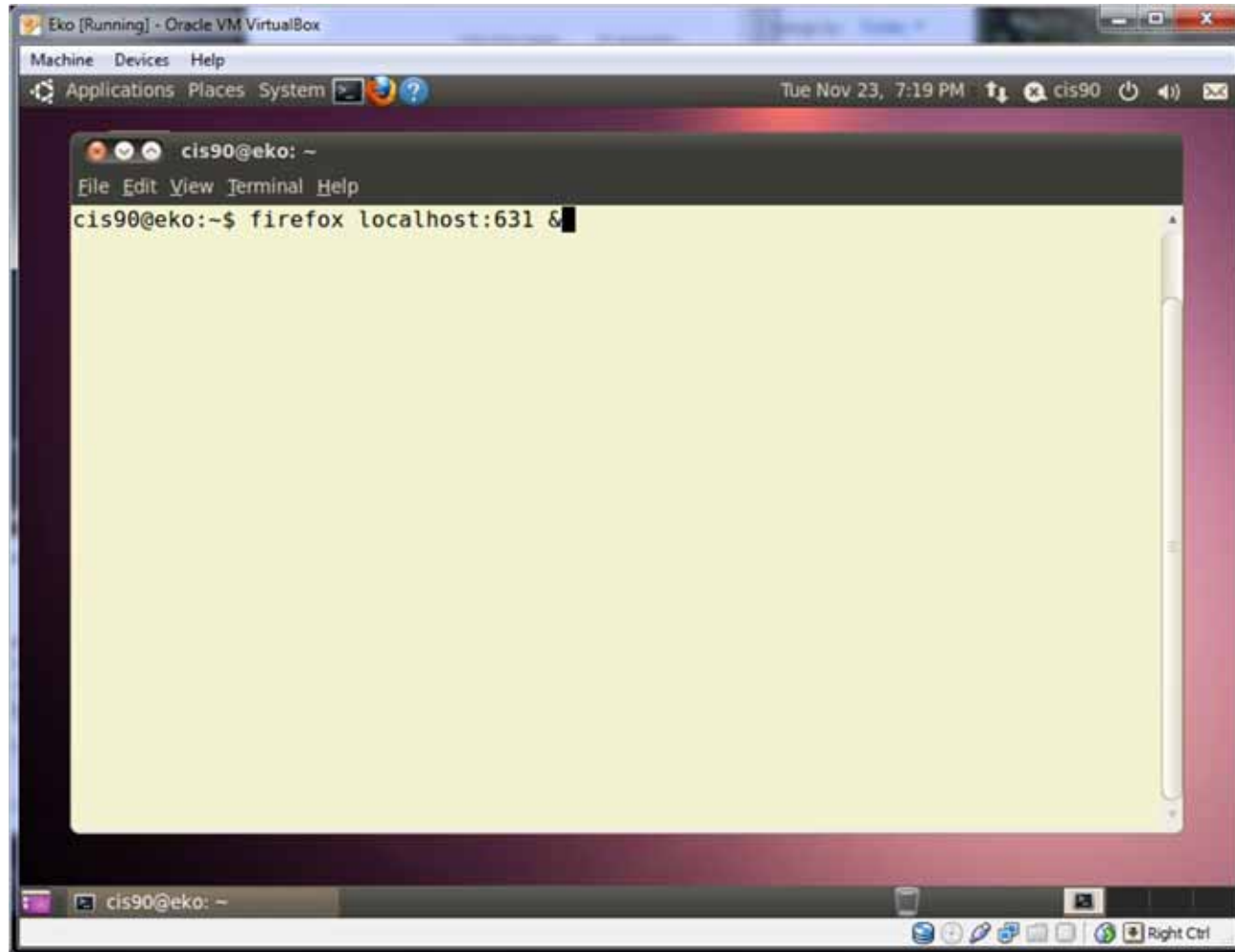
Login as cis90

CUPS



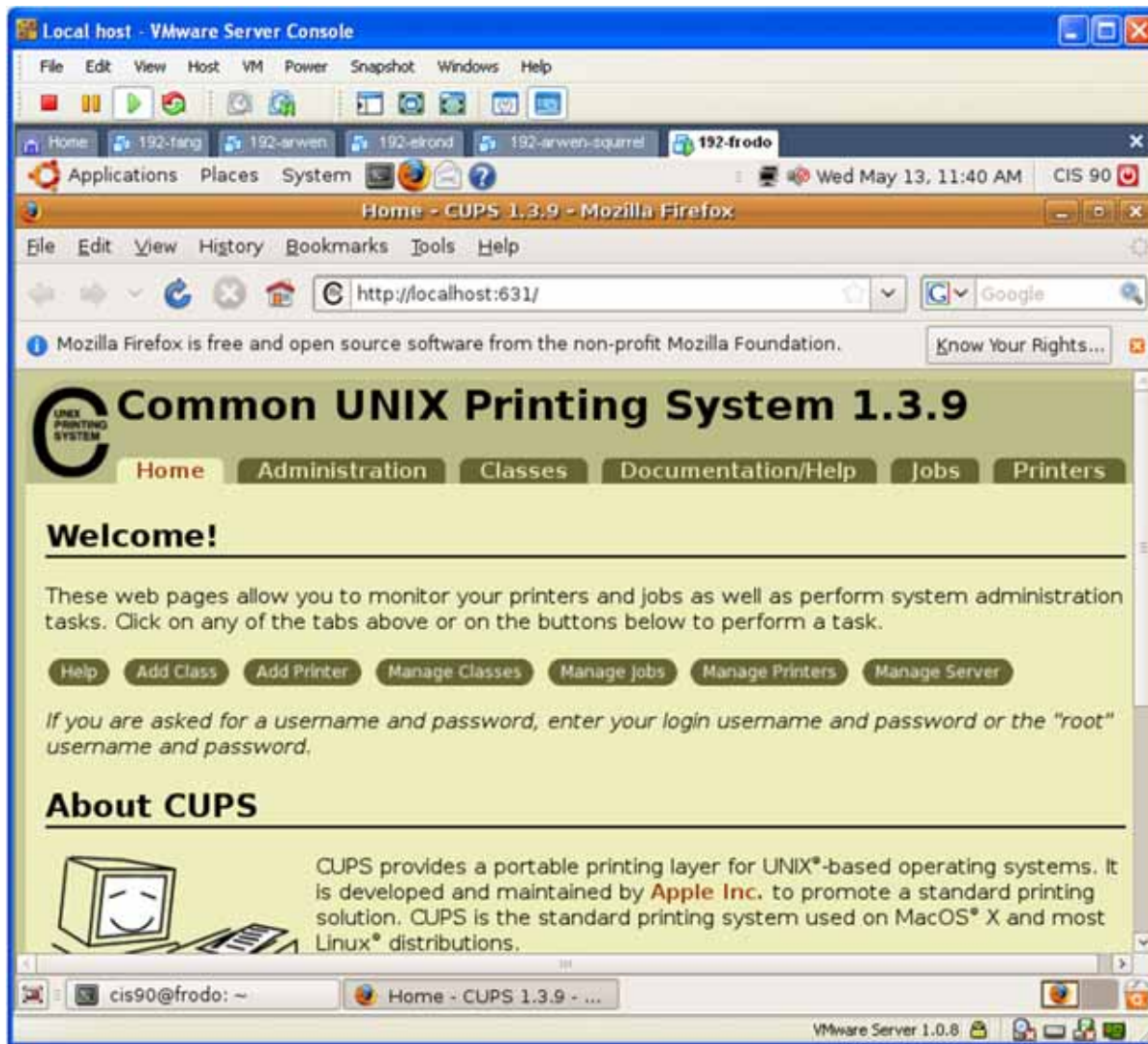
*Open the Eko
Terminal icon*

CUPS



Type the **firefox** command with **localhost:631** as the argument in the background with the **&**

CUPS



CUPS is managed by a web-based configuration utility on port 631

Local access only by default

CUPS

Next step is to add printers



*Printer: HP LaserJet 1320n
Connection: LAN*

CUPS

The LaserJets also have a web-based management utility



The screenshot shows a web browser window displaying the HP LaserJet 1320 series management utility. The browser address bar shows the URL `http://192.168.0.12/hp/device/`. The page title is "hp LaserJet 1320 series". The main content area is divided into several sections:

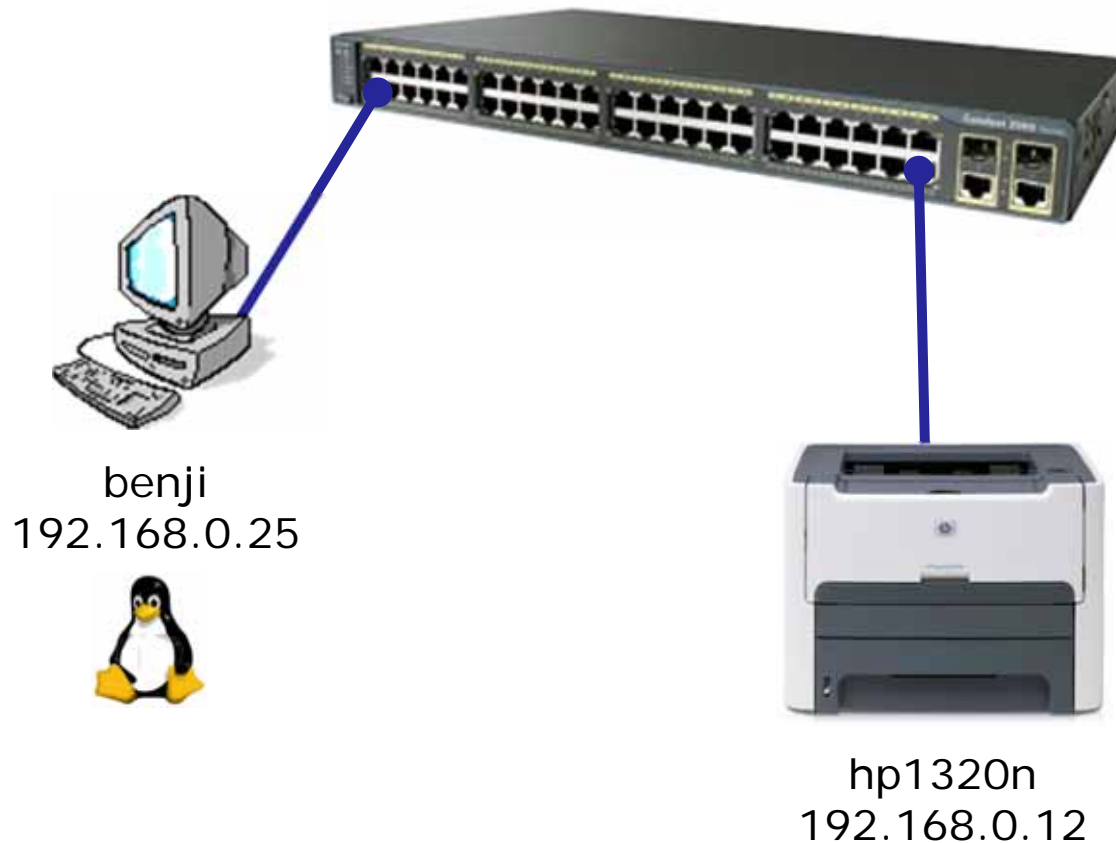
- Information** (selected):
 - Device Status
 - Configuration
 - Supplies Status
 - Event Log
 - Print Info Pages
- Other Links**:
 - Product Registration
 - Order Supplies
 - Product Support
- Device Status**:
 - Status: Ready
 - Buttons: Refresh Status, Enter, Cancel Job
- Supplies**:
 - Toner: (% Remaining)
 - Black Cartridge: 17%
 - Buttons: Supplies Details
- Product Information**:

Product Name:	hp LaserJet 1320 series
Formatter Number:	JH03T22
Product Serial Number:	CJHC6360LV
Service ID:	16101
Firmware Datecode:	20041024
Total Memory:	16 MBytes
Available Memory:	5.58 MBytes
IP Address:	192.168.0.12

*IP Address for this 1320n
is 192.168.0.12*

CUPS

This example will show how to add the HP 1320n as a networked printer.



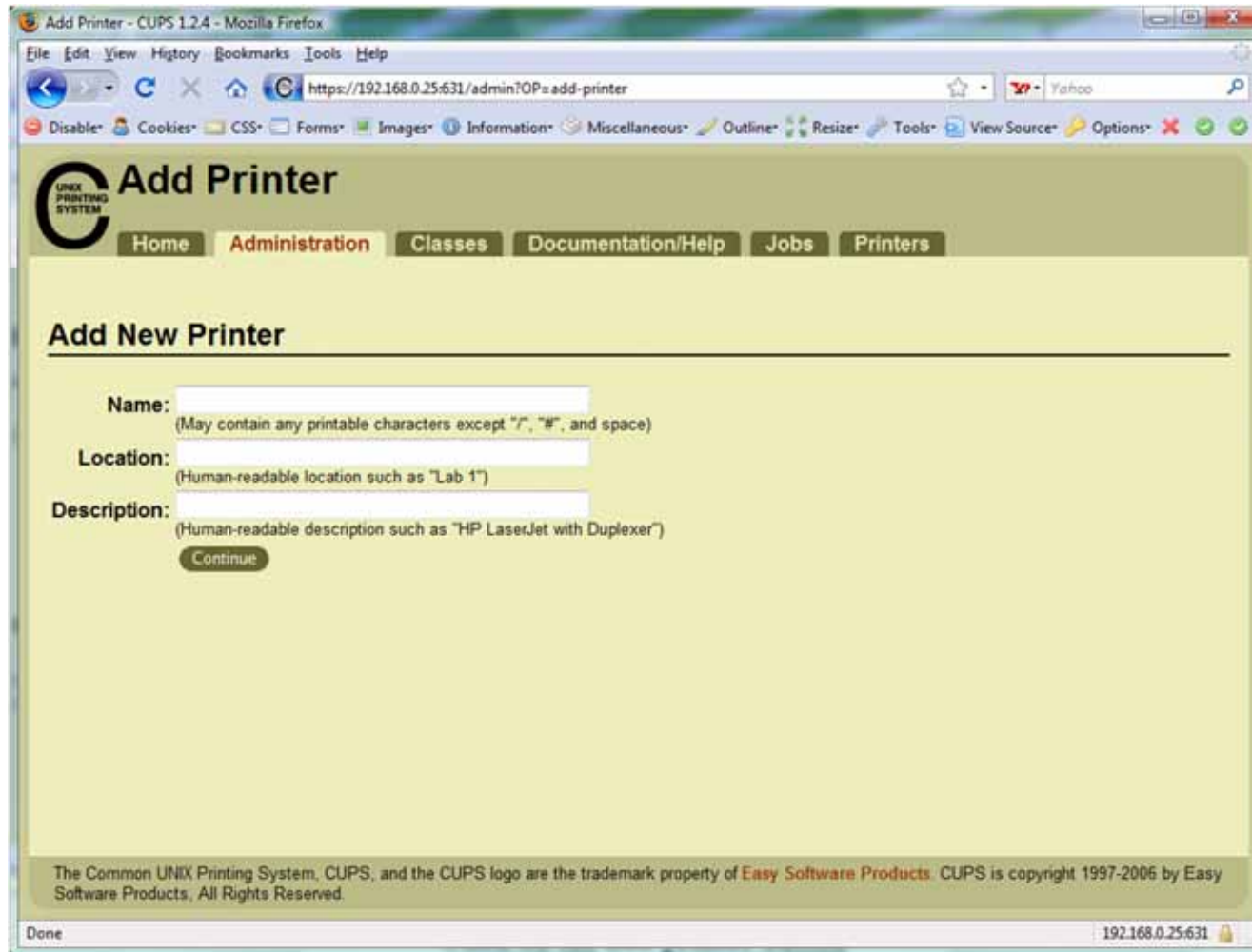
CUPS



To add in HP 1320N printer ...

... the first step is to click the Add Printer button

CUPS



*Now we can add
the LaserJet*

CUPS

Add Printer - CUPS 1.2.4 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.0.25:631/admin?OP=add-printer

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Add Printer

Home Administration Classes Documentation/Help Jobs Printers

Add New Printer

Name: LaserJet
(May contain any printable characters except "/", "#", and space)

Location: Family Room
(Human-readable location such as "Lab 1")

Description: HP LaserJet 1320 PCL 5e
(Human-readable description such as "HP LaserJet with Duplexer")

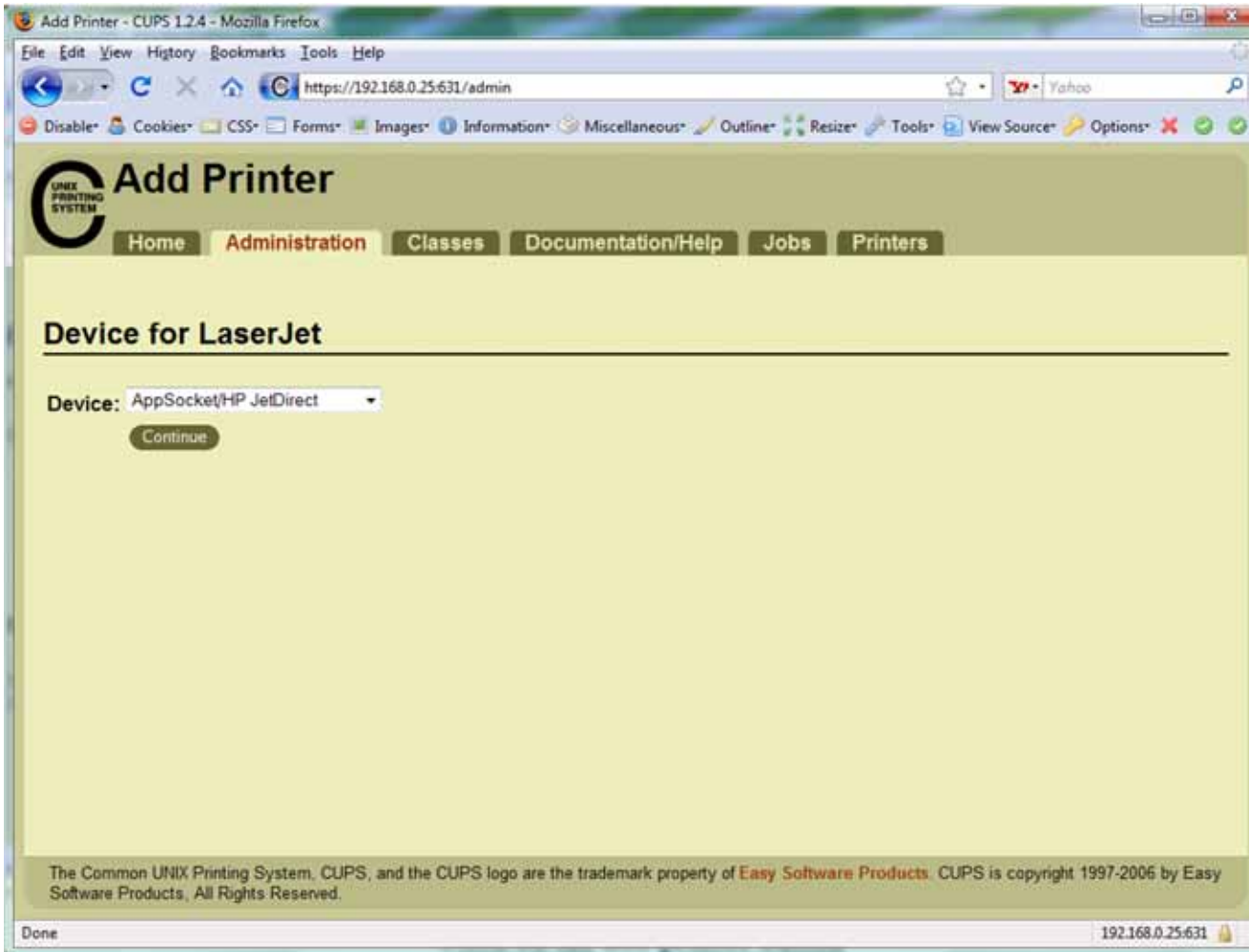
Continue

The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. CUPS is copyright 1997-2006 by Easy Software Products. All Rights Reserved.

Done 192.168.0.25:631

Fill in basic information

CUPS

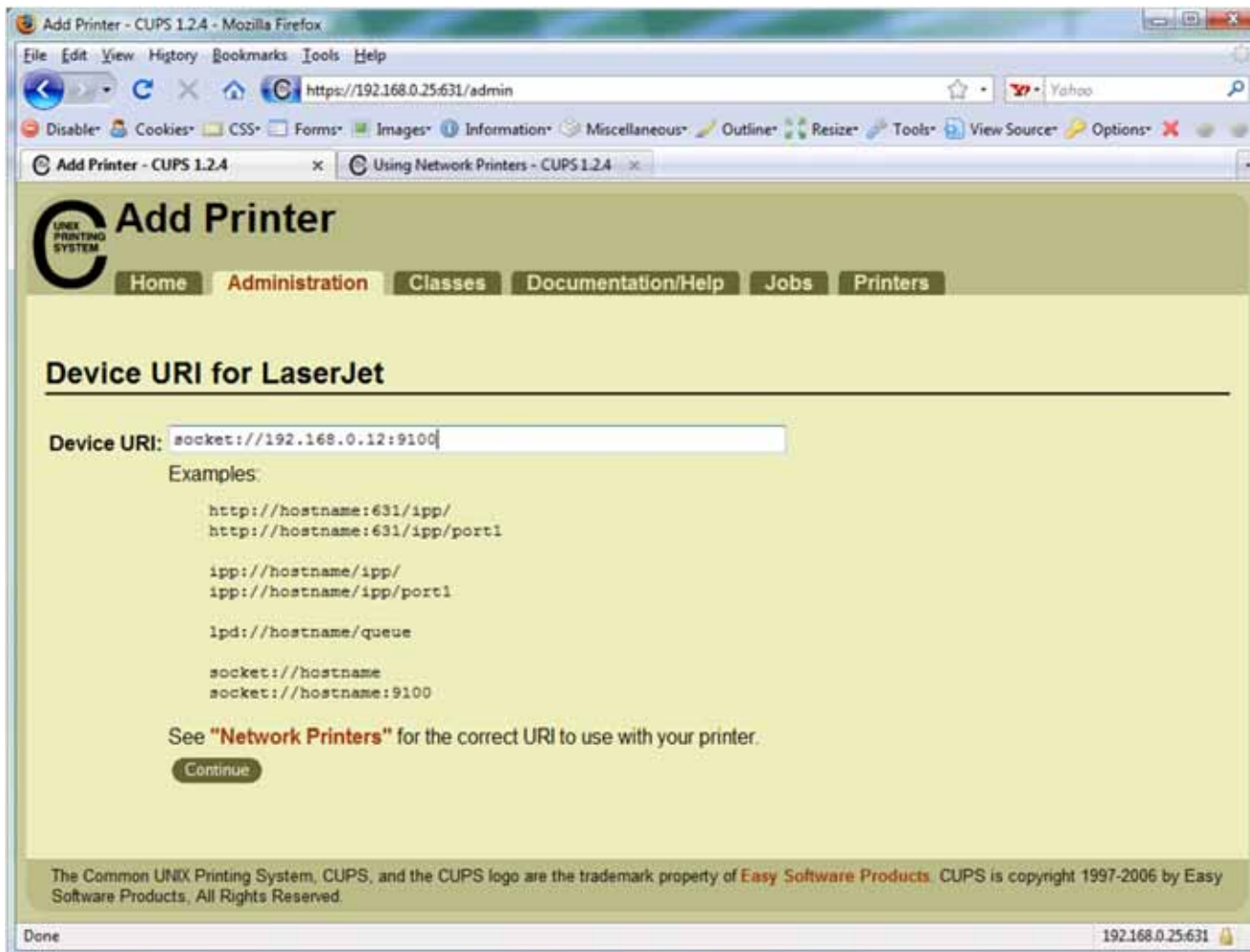


*We will use
JetDirect.*

*JetDirect is a
small printer
server built into
some of HP's
printers.*

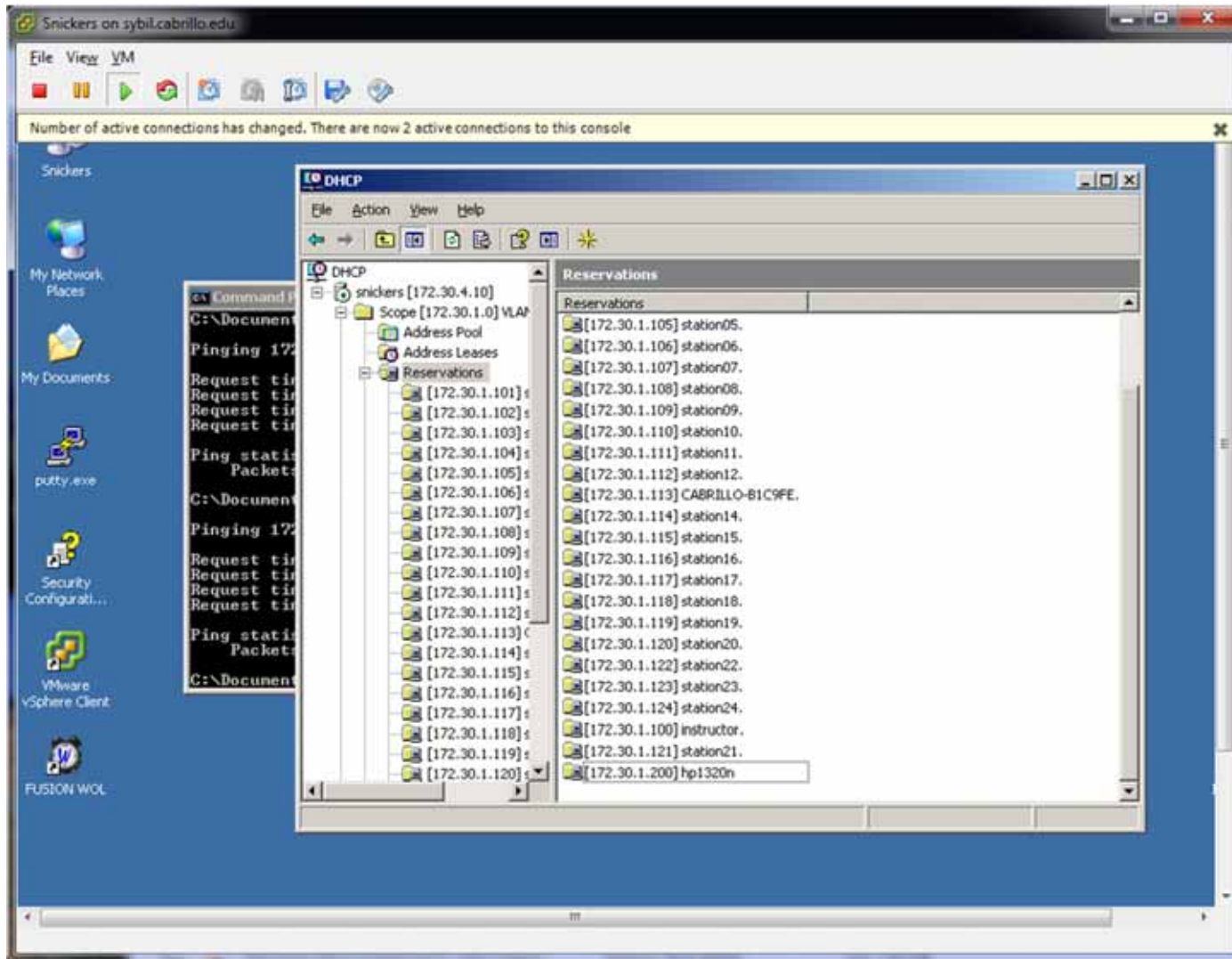
CUPS

socket://192.168.0.12:9100



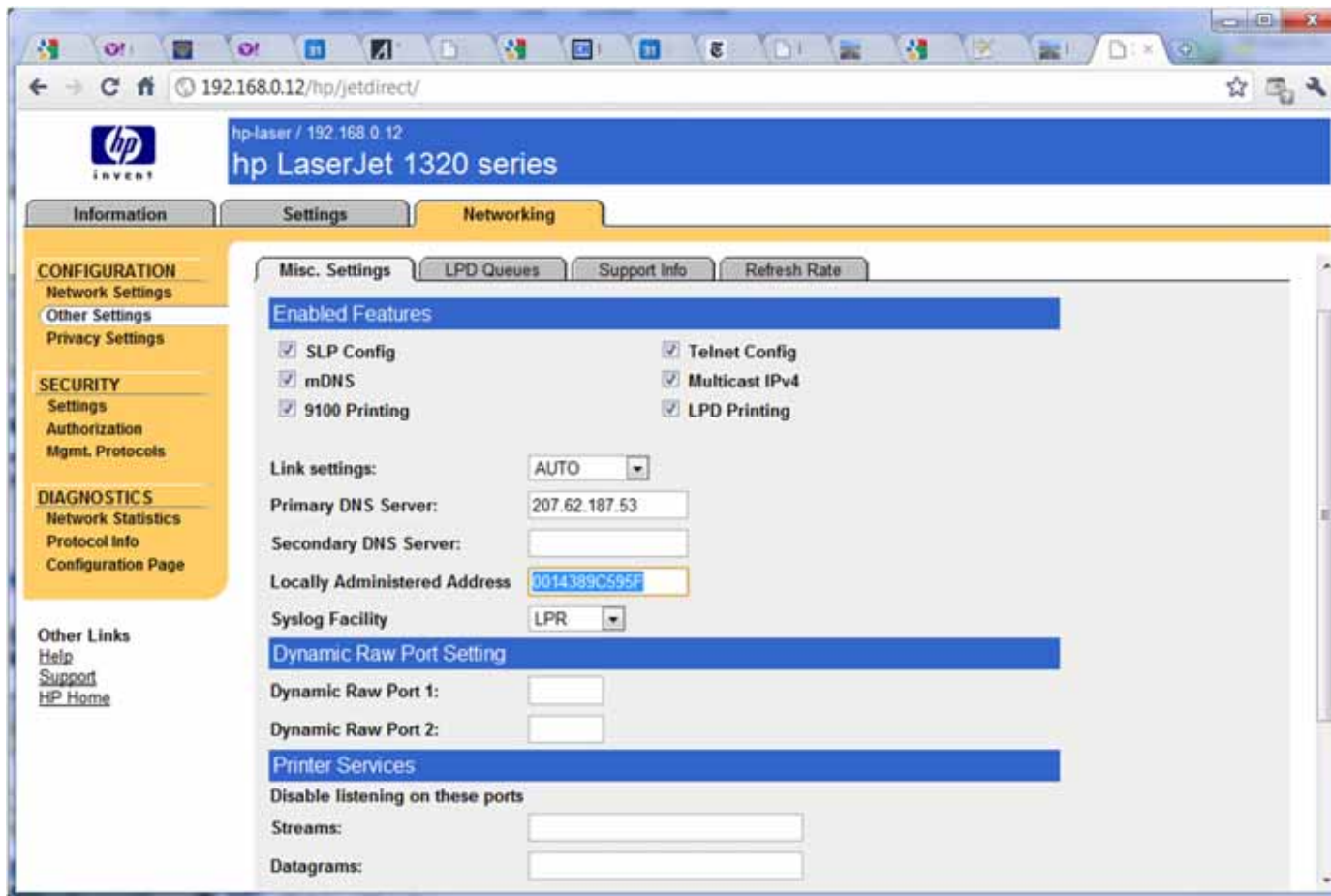
*This defines
how to
communicate
with the
printer*

CUPS



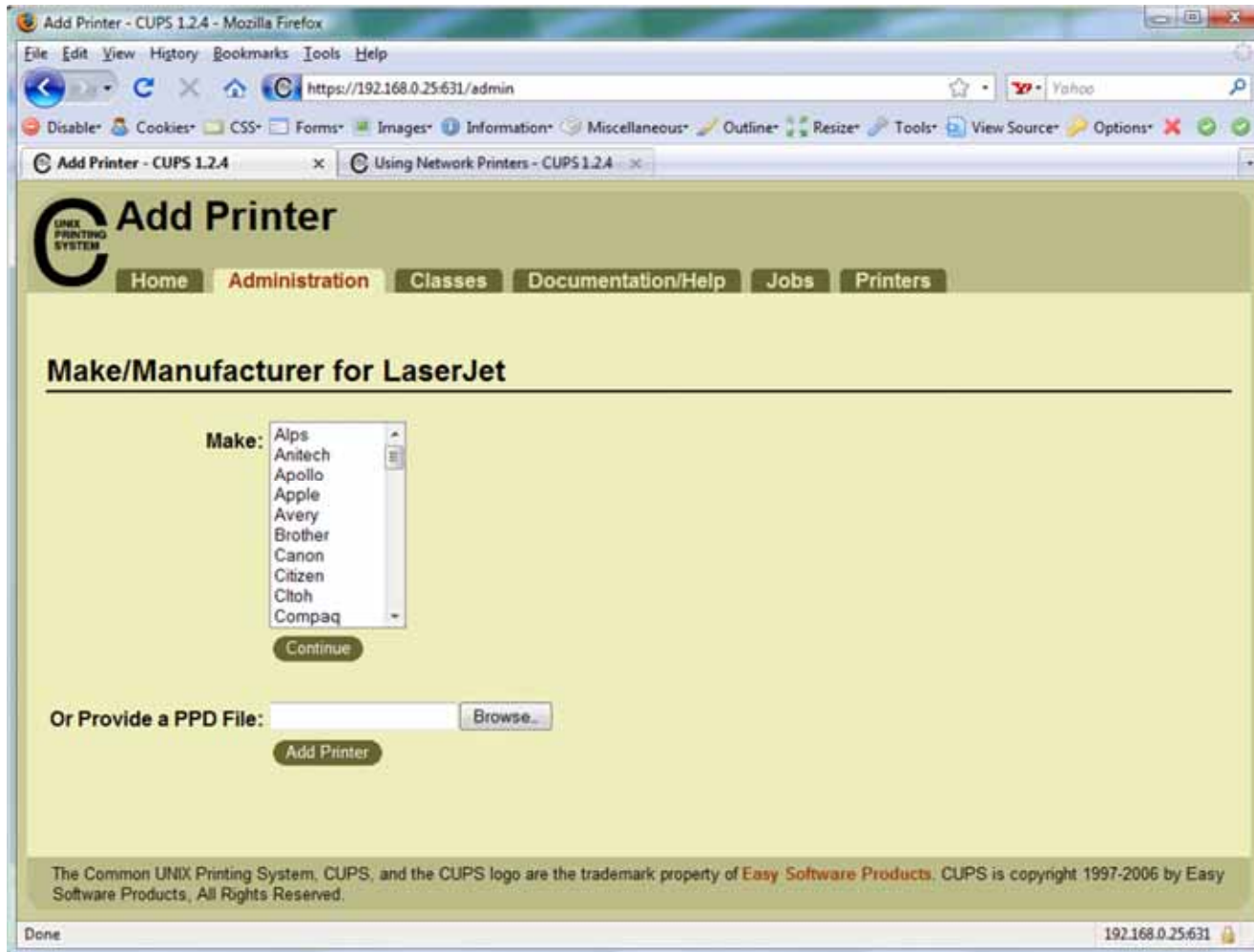
Room 2501: 172.30.1.200 for 0014389C595F

CUPS



Room 2501: 172.30.1.200 for 0014389C595F

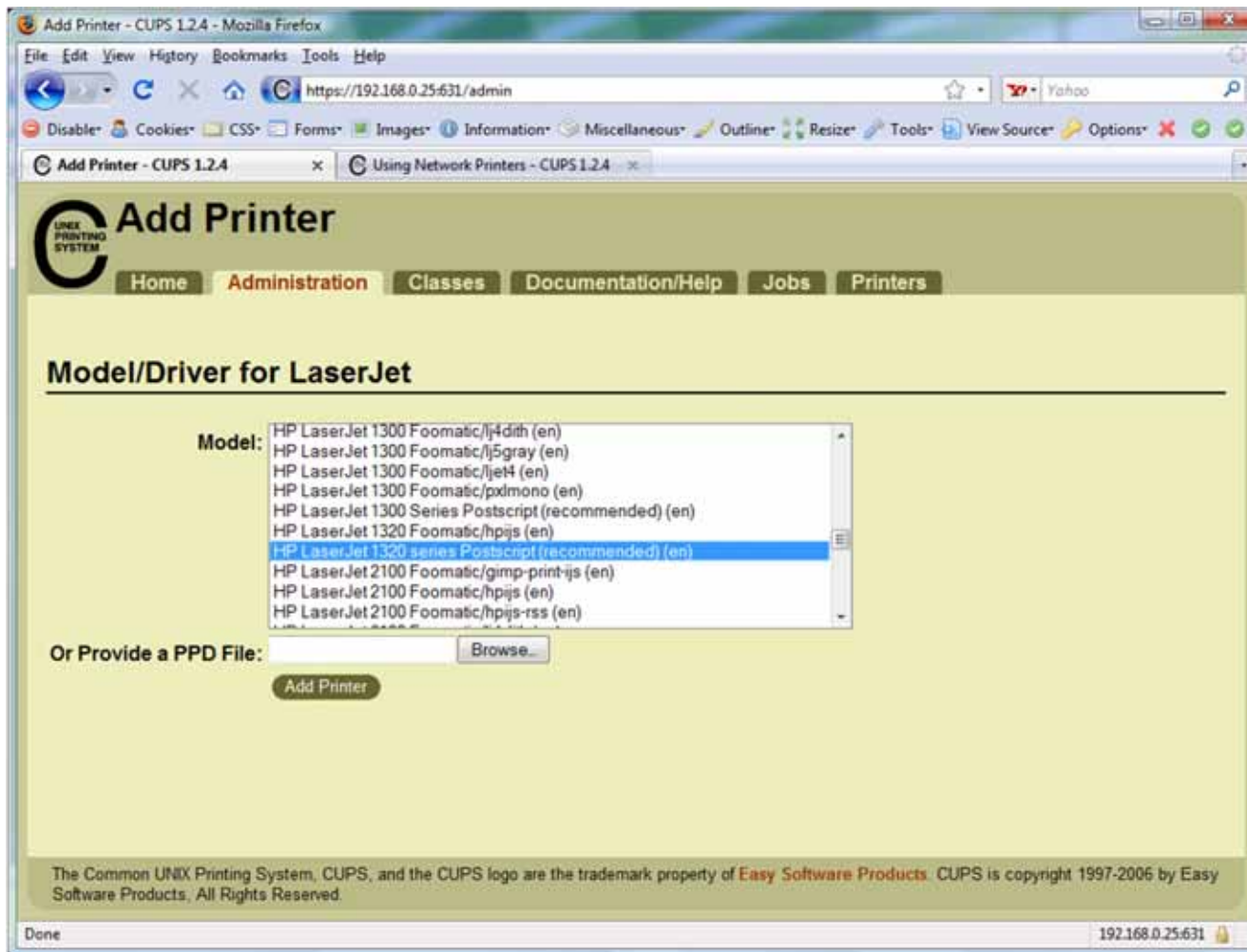
CUPS



(you will need to enter root's password)

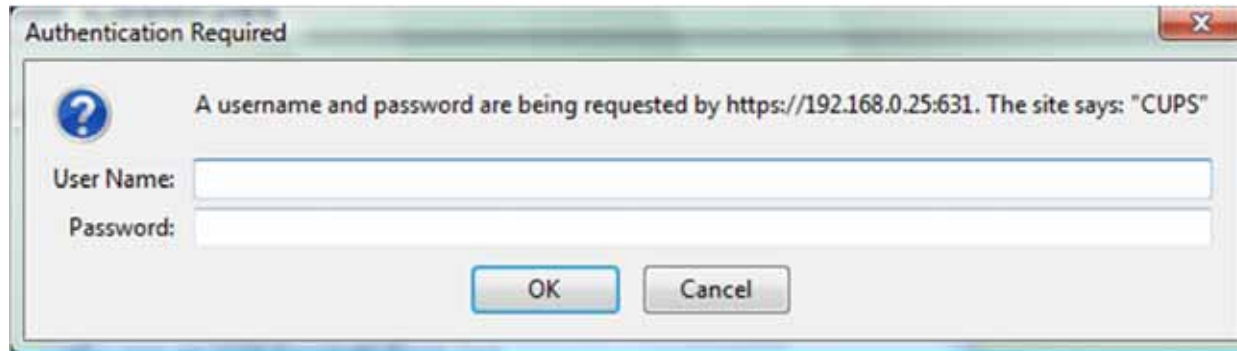
Service will restart

CUPS



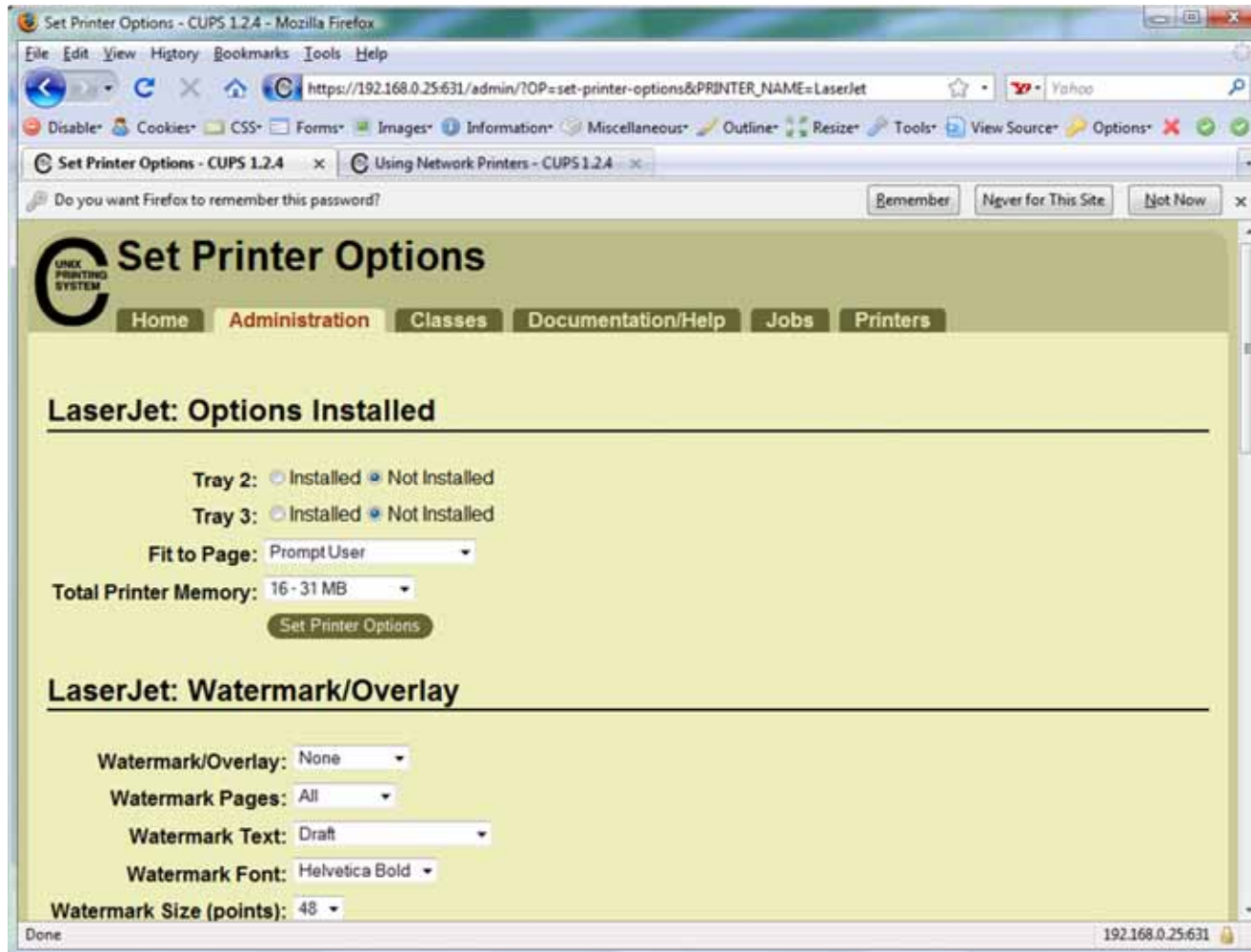
We will choose hp LaserJet 1320 series Postscript (recommended) (en)

CUPS



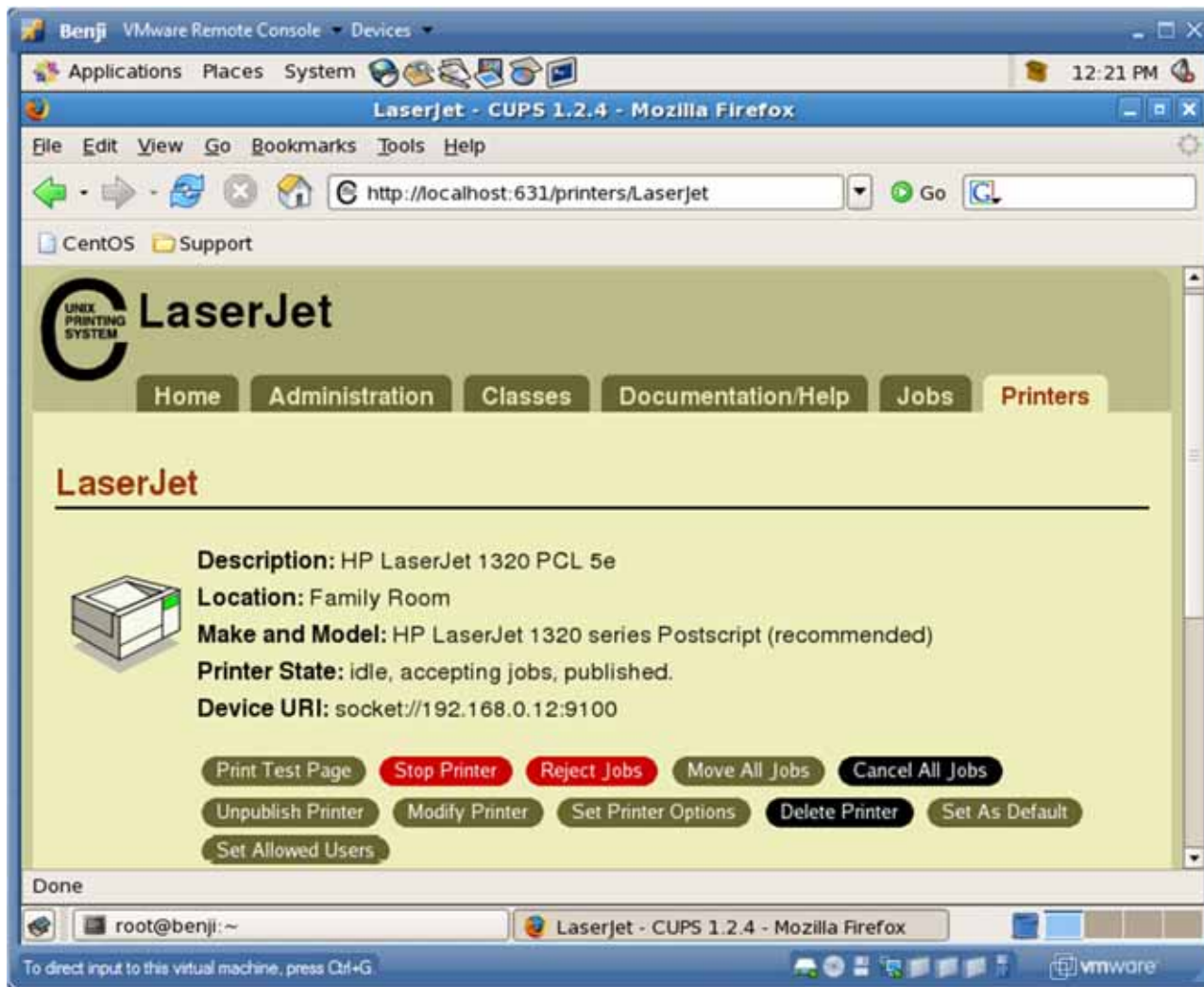
To finally add the printer it will be necessary authenticate as root

CUPS



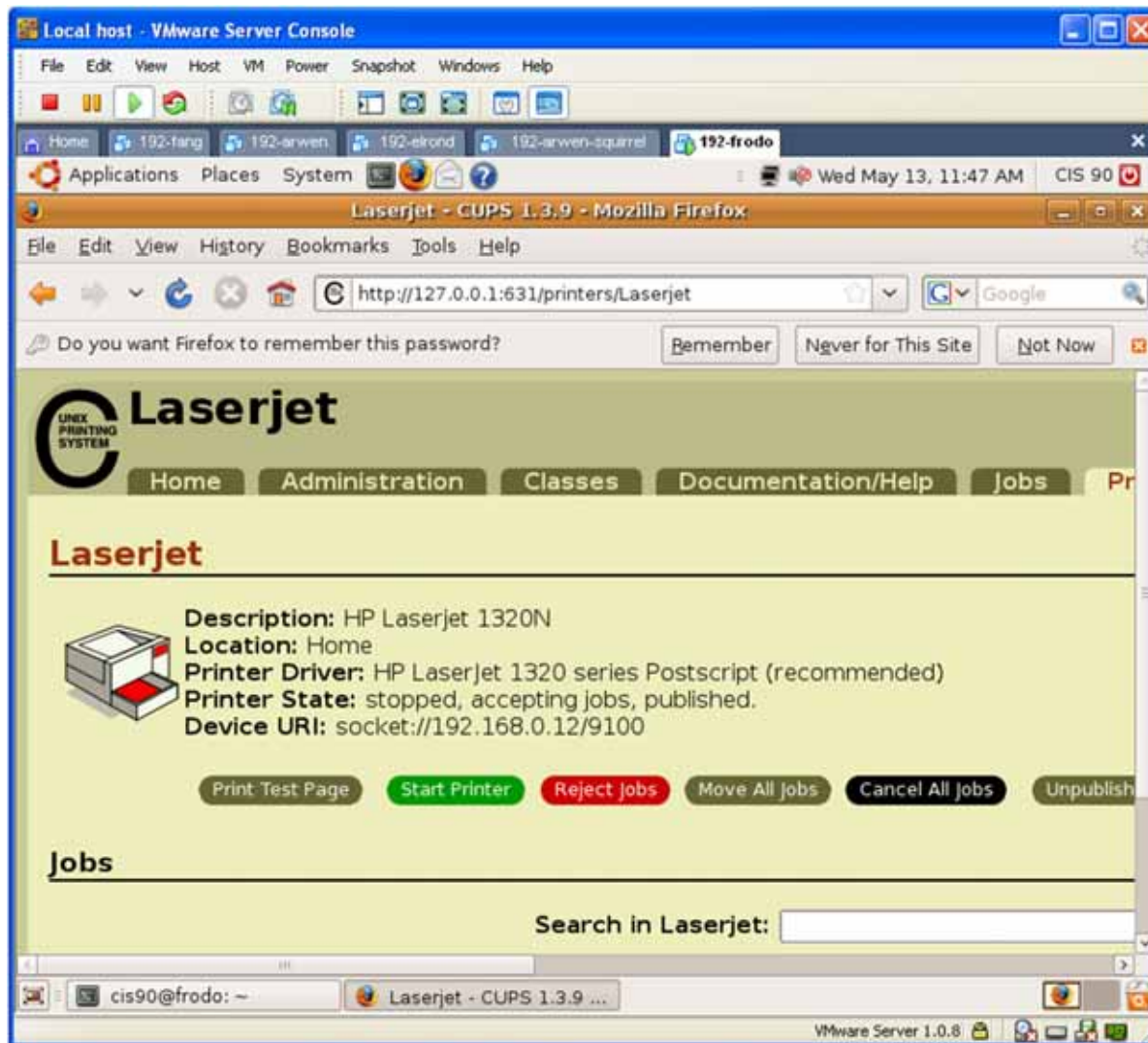
*Printer has
been added*

CUPS



View of newly added printer from Printer tab

CUPS



Configure the printer so it is stopped but still accepts print jobs

CUPS



Lets add second printer



*Printer: hp photosmart 7550 (color inkjet technology)
Connection: USB*

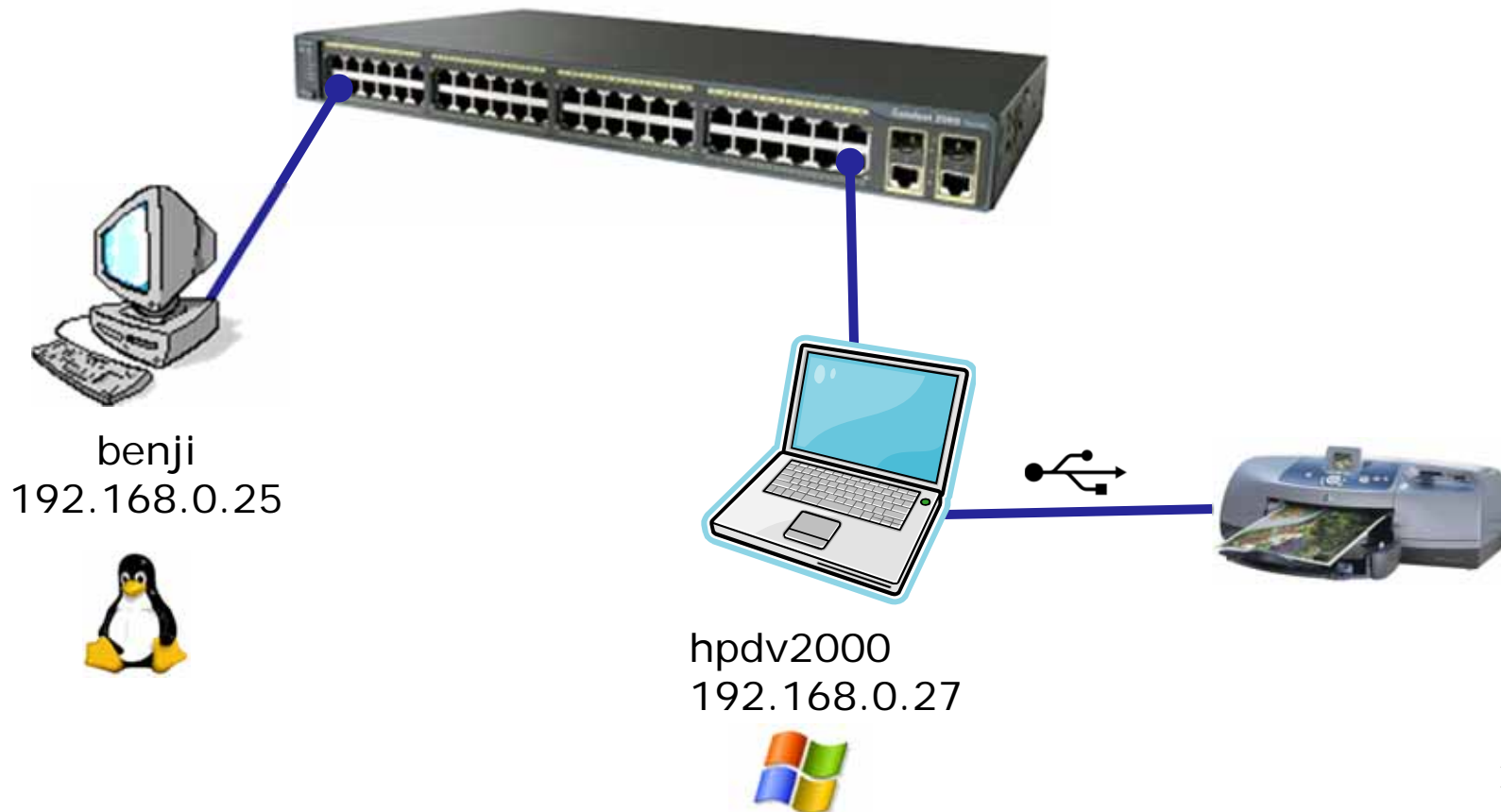
Sidetrack – The previous 7550 "Hot Lips"



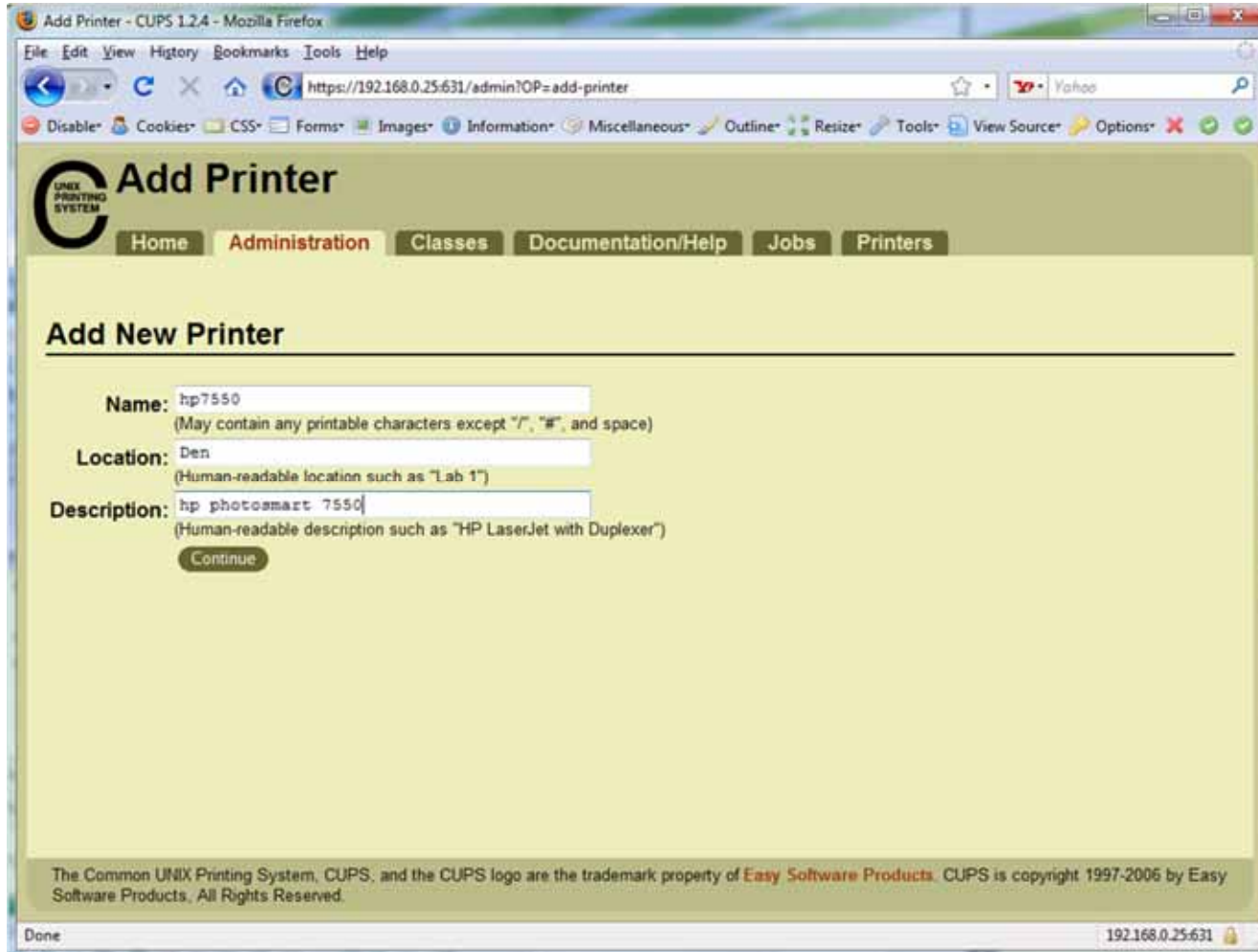
*6 G's of acceleration
8-pen turret
Grit wheel technology from HP Labs*

CUPS

The second printer is connected by USB to a Windows notebook computer



CUPS



First step is the same which is to fill out basic information on printer

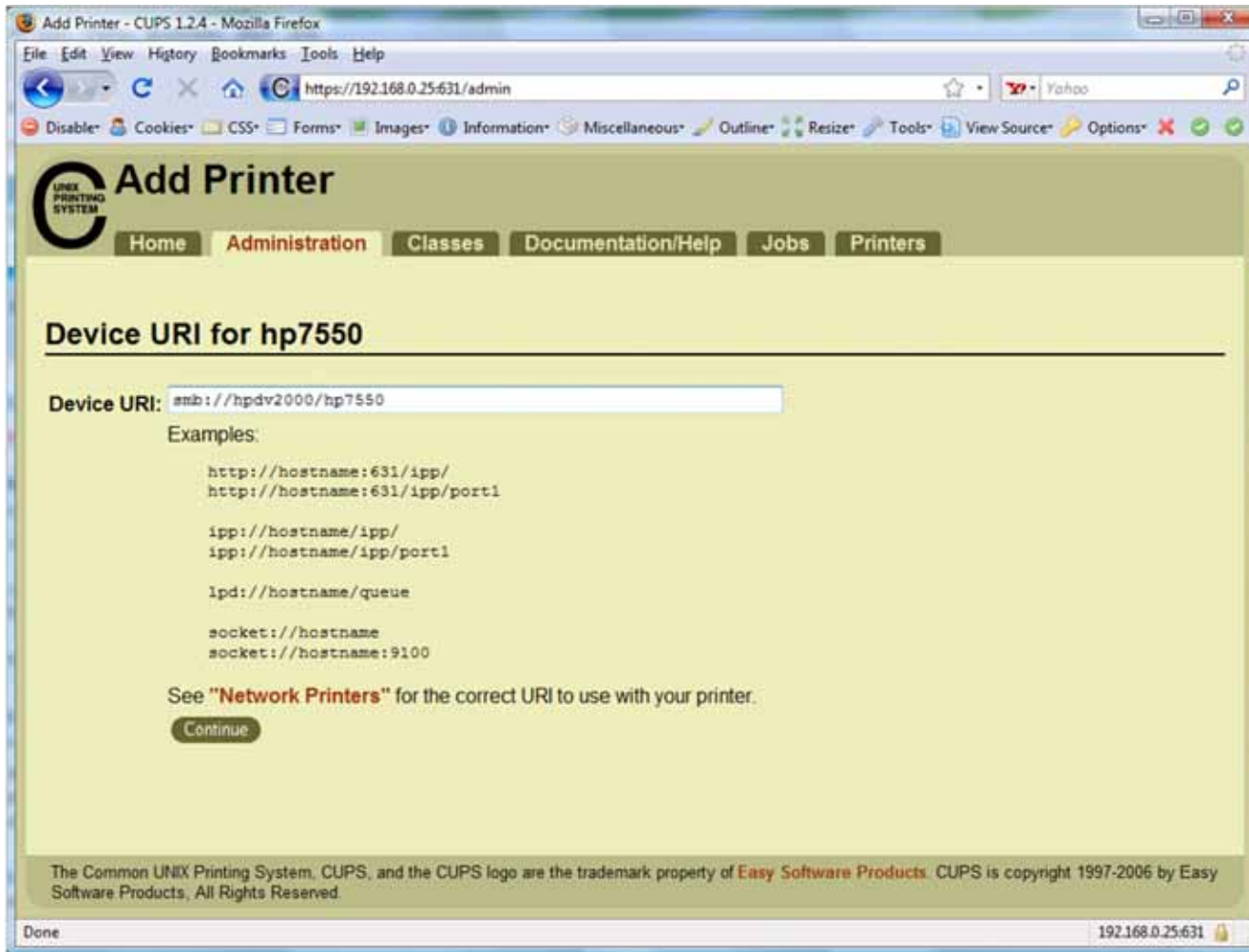
CUPS



For this connection we will use Samba. Samba implements Windows file and print services sharing on Linux.

Note Windows uses SMB (Server Message Block) protocol to implement these services

CUPS



Will need to specify the Windows print share

CUPS

Will need to specify the Windows print share as //hostname/printsharename

The image shows two overlapping windows. The background window is the CUPS 1.2.4 web interface in Mozilla Firefox, titled "Add Printer - CUPS 1.2.4 - Mozilla Firefox". The address bar shows "https://192.168.0.25:631/admin". The page title is "Add Printer" and the main heading is "Device URI for hp7550". The "Device URI:" field contains "smb://hpdv2000/hp7550". Below this, there are examples of URIs for http, ipp, lpd, and socket. A "Continue" button is visible at the bottom of the form.

The foreground window is a Windows command prompt titled "Administrator: C:\Windows\system32\cmd.exe". It shows the following commands and output:

```
C:\Users\Administrator>hostname  
hpdv2000  
C:\Users\Administrator>net share
```

Share name	Resource	Remark
C\$	C:\	Default share
D\$	D:\	Default share
J\$	J:\	Default share
print\$	C:\Windows\system32\spool\drivers	Printer Drivers
IPC\$		Remote IPC
ADMIN\$	C:\Windows	Remote Admin
hp LaserJet 1320 PCL 5	192.168.0.12	Spooled hp LaserJet 1320 PCL 5e
hp7550	DOT4_001	Spooled hp7550

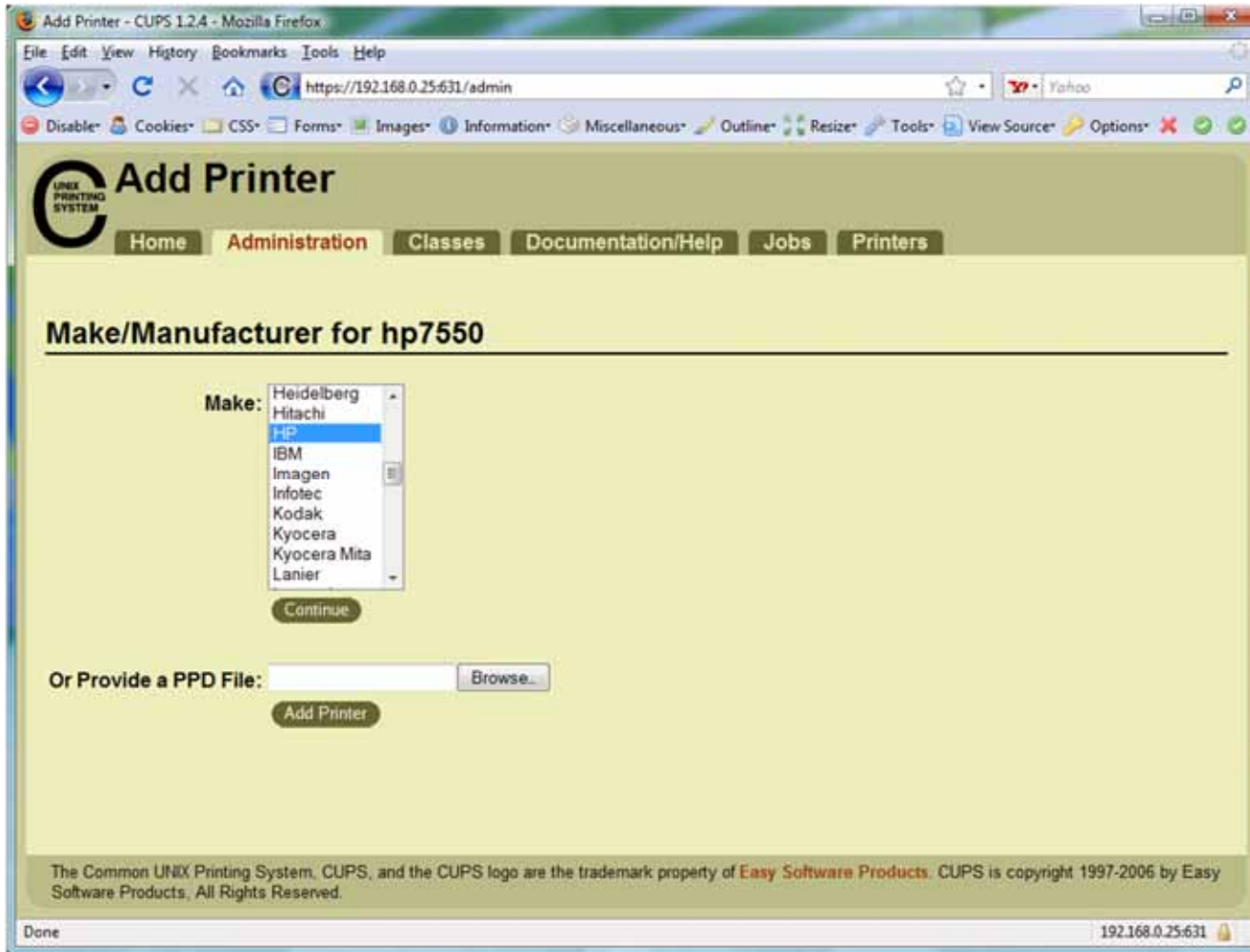
The command prompt concludes with "The command completed successfully." and the prompt "C:\Users\Administrator>".

CUPS

Ways to specify a Windows share

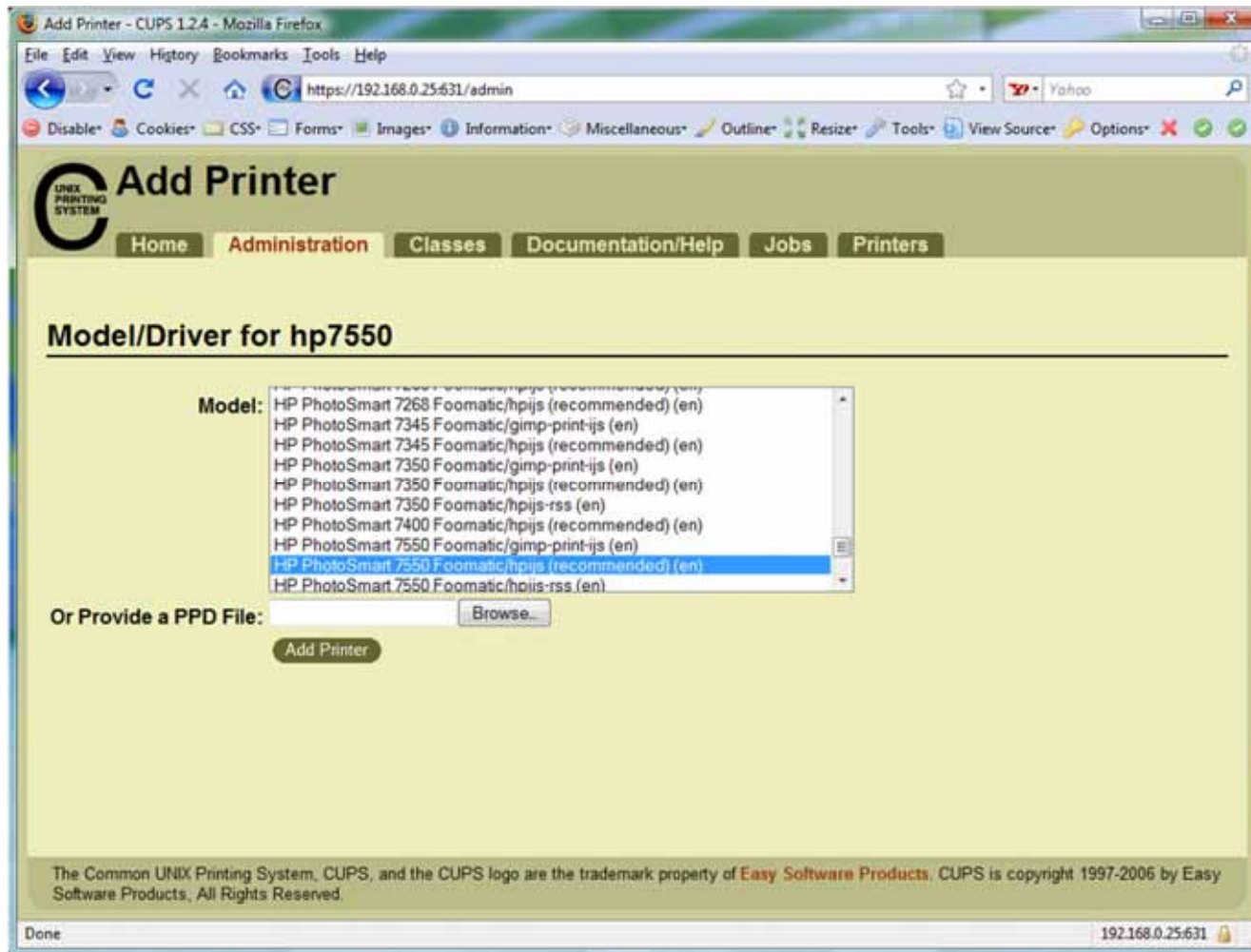
	Username and password Not required
This machine is in the same workgroup	<code>smb://server/sharename</code>
This machine is in a different workgroup	<code>smb://workgroup/server/sharename</code>
	Username and password required
This machine is in the same workgroup	<code>smb://username:password@server/sharename</code>
This machine is in a different workgroup	<code>smb://username:password@workgroup/server/sharename</code>

CUPS



*Select make
of printer*

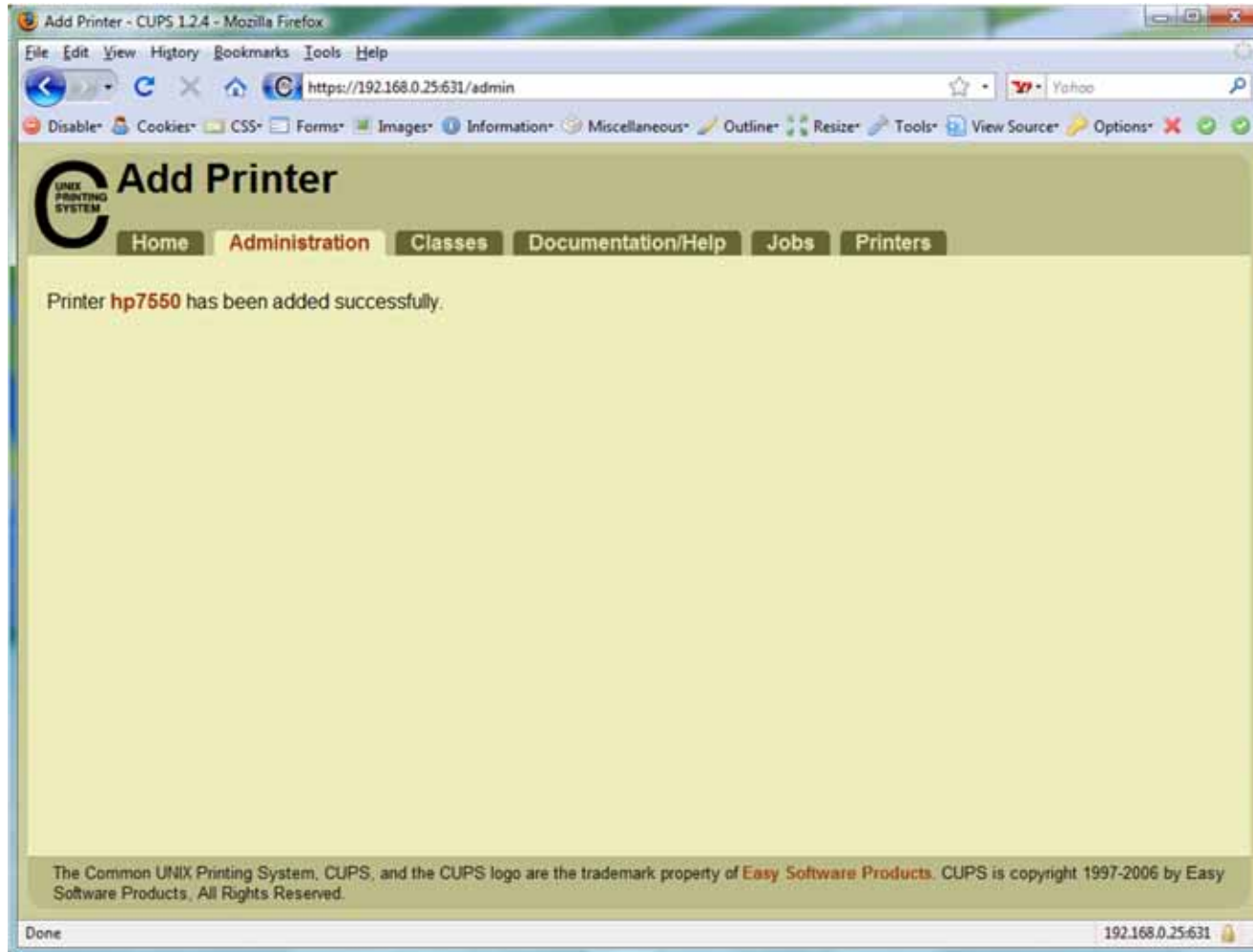
CUPS



*Select model of
printer*

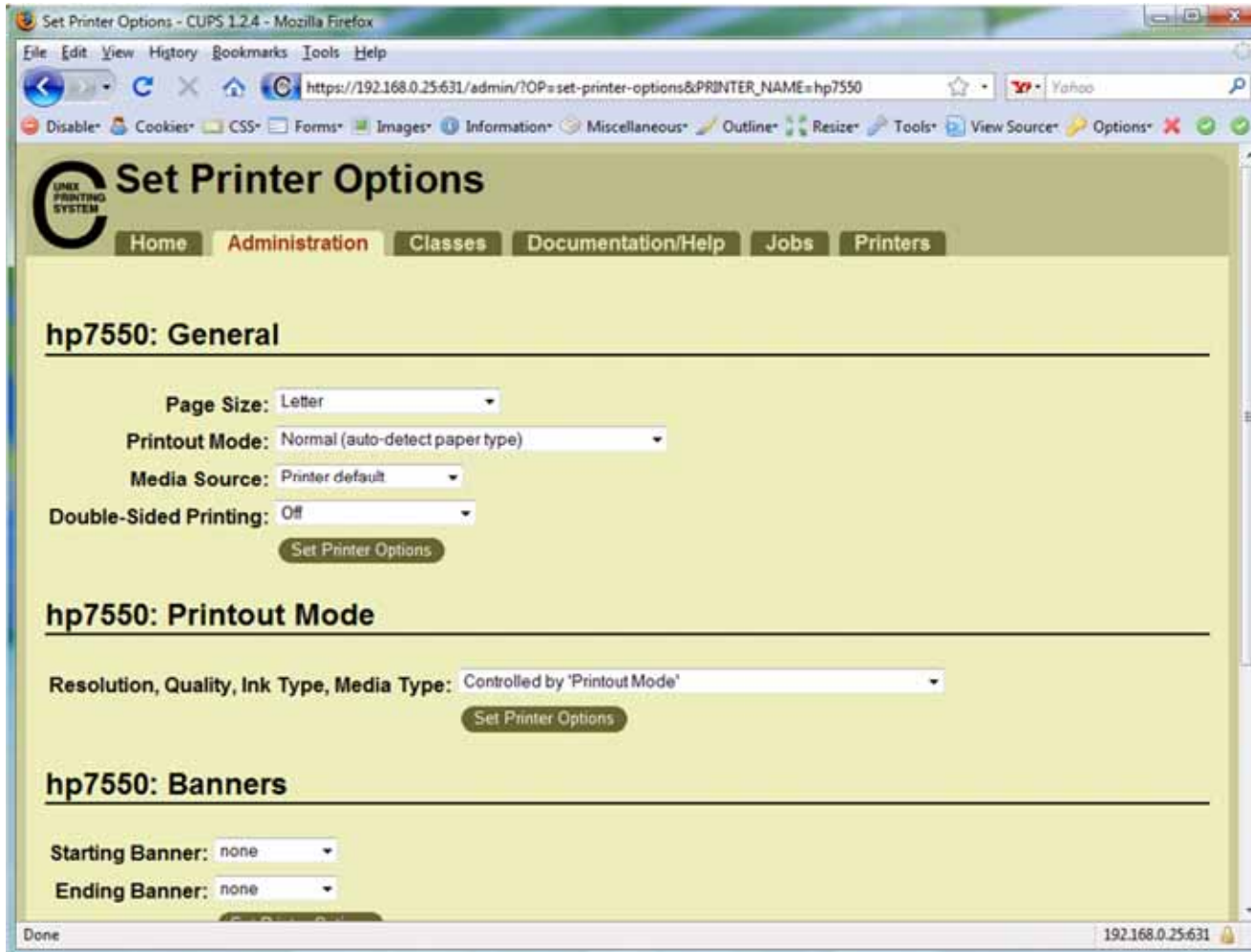
*HP PhotoSmart 7550
Foomatic/hpijs
(recommended) (en)*

CUPS



Printer has been added

CUPS



View and set options as needed

Before using the printer we need to check that SAMBA is installed



Newer CUPS

Add Printer - CUPS 1.4.3 - Mozilla Firefox
File Edit View History Bookmarks Tools Help

http://localhost:631/admin/

Most Visited Getting Started Latest Headlines

Add Printer - CUPS 1.4.3

Do you want Firefox to remember the password for "root" on http://localhost:631? Remember Never for This Site Not Now

Home Administration Classes Online Help Jobs Printers

Add Printer

Local Printers:

- SCSI Printer
- HP Printer (HPLIP)
- HP Fax (HPLIP)

Discovered Network Printers:

- hp LaserJet 1320 series (9C595F) (hp hp LaserJet 1320 series)
- hp LaserJet 1320 series (9C595F) (hp hp LaserJet 1320 series)
- Photosmart Prem-Web C309n-s [012D30] (HP HP Photosmart Prem-Web C309n-s)
- hp LaserJet 1320 series (hp LaserJet 1320 series)
- Photosmart Prem-Web C309n (HP Photosmart Prem-Web C309n-s)

Other Network Printers:

- Windows Printer via SAMBA
- LPD/LPR Host or Printer
- AppSocket/HP JetDirect
- Internet Printing Protocol (http)
- Internet Printing Protocol (ipp)
- Backend Error Handler

Continue

Done



Add Printer

Name:
(May contain any printable characters except "/", "#", and space)

Description:
(Human-readable description such as "HP LaserJet with Duplexer")

Location:
(Human-readable location such as "Lab 1")

Connection: socket://192.168.0.12

Sharing: Share This Printer

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

Add Printer

Name: hp_LaserJet_1320_series
Description: hp LaserJet 1320 series
Location: Family room
Connection: socket://192.168.0.12
Sharing: Do Not Share This Printer

Make: HP

Model: HP LaserJet 1320 Series hpijs pci3, 3.10.2 (en)
HP LaserJet 1320 series Postscript (recommended) (en, da, de, es, fi, fr, it, ja, ko, nl, nb, pt, sv, zh_CN, zh_TW)
HP 910 hpijs, 3.10.2 (en)
HP 915 hpijs, 3.10.2 (en)
HP 2000C Foomatic/pci3 (en)
HP 2000c hpijs, 3.10.2 (en)
HP 2500C Foomatic/pci3 (en)
HP 2500c hpijs, 3.10.2 (en)
HP 2500CM Foomatic/Postscript (recommended) (en)
HP 2563 Foomatic/lp2563 (recommended) (en)

Or Provide a PPD File:

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

The screenshot shows a Mozilla Firefox browser window titled "Set Printer Options - CUPS 1.4.3". The address bar shows the URL "http://localhost:631/admin". The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. Below the address bar are navigation buttons (back, forward, home, refresh) and a search bar with the Google logo. The browser's tab bar shows a single tab titled "Set Printer Options - CUPS 1.4.3".

The main content area of the browser displays the CUPS administration interface. At the top, there is a navigation menu with buttons for Home, Administration, Classes, Online Help, Jobs, and Printers. The main heading is "Set Default Options for hp_LaserJet_1320_series". Below this heading is a button labeled "Query Printer for Default Options".

Underneath, there is a horizontal menu with tabs for "Options Installed", "General", "Watermark/Overlay", "Image Quality", "Banners", "Policies", and "Port Monitor". The "Options Installed" tab is currently selected.

The "Options Installed" section contains the following settings:

- Tray 2:** Installed Not Installed
- Tray 3:** Installed Not Installed
- Fit to Page:** Prompt User (dropdown menu)
- Total Printer Memory:** 16 - 31 MB (dropdown menu)

At the bottom of the settings section is a button labeled "Set Default Options".

The browser's status bar at the bottom left shows the word "Done".

Set Printer Options - CUPS 1.4.3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:631/admin

Most Visited Getting Started Latest Headlines

Loading...

Home Administration Classes Online Help Jobs Printers

Set Default Options for hp_LaserJet_1320_series

Printer [hp_LaserJet_1320_series](#) default options have been set successfully.

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Waiting for localhost...

hp_LaserJet_1320_series (Idle, Accepting Jobs, Not Shared)

Maintenance Administration

Description: hp LaserJet 1320 series
Location: Family room
Driver: HP LaserJet 1320 series Postscript (recommended) (grayscale, 2-sided printing)
Connection: socket://192.168.0.12
Defaults: job-sheets=none, none media=na_letter_8.5x11in sides=one-sided

Jobs

Search in hp_LaserJet_1320_series: Search Clear

Show Completed Jobs Show All Jobs

No jobs.

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

Use this pull down menu to select this printer as default

Printers - CUPS 1.4.3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:631/printers/

Printers - CUPS 1.4.3

Home Administration Classes Online Help Jobs Printers

Search in Printers: Search Clear

Showing 1 of 1 printer.

Queue Name	Description	Location	Make and Model	Status
hp_LaserJet_1320_series	hp LaserJet 1320 series	Family room	HP LaserJet 1320 series Postscript (recommended)	Idle

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

The image shows a terminal window on the left and a printer control interface on the right. The terminal window has a title bar with window control icons and the text 'cis90@eko: ~'. Below the title bar is a menu bar with 'File Edit View Terminal Help'. The terminal content shows the following commands and output:

```
cis90@eko:~$ echo print me please | lp  
request id is hp_LaserJet_1320_series-1 (0 file(s))  
cis90@eko:~$
```

The printer control interface on the right has a title bar with a search icon, a Google logo, and the text 'Google'. Below the title bar is a search bar with the text 'Printers' and a search button. Below the search bar are 'Search' and 'Clear' buttons. Below these buttons is a 'Control' section with 'Cancel Job' and 'Move Job' buttons. At the bottom of the interface is a footer with the text 'CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.' and the word 'Done'.

Jobs - CUPS 1.4.3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:631/jobs/

Document printed
Document '(stdin)' has been sent to 'hp_LaserJet_1320_series' for printing

Jobs - CUPS 1.4.3

Home Administration Classes Online Help Jobs Printers

Search In Jobs: Search Clear

Show Completed Jobs Show All Jobs

Showing 1 of 1 active job.

ID	Name	User	Size	Pages	State	Control
hp_LaserJet_1320_series-1	(stdin)	cis90	1k	1	processing since Thu 12 May 2011 11:00:32 AM PDT	Cancel Job Move Job

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

http://localhost:631/jobs/

Printing in Linux

Printing

System V based print subsystem

- lp (to print)
- lpstat (queue management)
- cancel (to remove jobs)

BSD based print subsystem

- lpr (to print)
- lpq (queue management)
- lprm (to remove jobs)

CUPS

- Provides both System V and Berkeley based command-line interfaces
- Supports new Internet Printing Protocol
- Works with Samba

We will be just looking at CUPS

CUPS

lpstat command

On the Frodo VM



```
File Edit View Terminal Tabs Help
cis90@frodo: ~
cis90@frodo:~$ lpstat
cis90@frodo:~$ lpstat -p
printer Laserjet disabled since Wed 13 May 2009 11:46:56 AM PDT -
    reason unknown
cis90@frodo:~$ lpstat -p -d
printer Laserjet disabled since Wed 13 May 2009 11:46:56 AM PDT -
    reason unknown
no system default destination
cis90@frodo:~$ █
```

*The -p option will show the available printers
The -d option will identify the default printer*

CUPS

lpstat command



On Opus

```
/home/cis90/roddyduk $ lpstat -p -d  
printer epson disabled since Tue 11 Nov 2008 01:36:13 PM PST -  
    reason unknown  
printer hplaser disabled since Tue 11 Nov 2008 01:36:13 PM PST -  
    reason unknown  
system default destination: hplaser  
/home/cis90/roddyduk $
```

The -p option will show the available printers
The -d option will identify the default printer

CUPS

lp and lpr commands



```
/home/cis90/roddyduk $ lp -d hplaser lab10  
request id is hplaser-3 (1 file(s))
```

*Either command will print
lab10 to the selected printer*

```
/home/cis90/roddyduk $ lpr -P hplaser lab10
```

```
/home/cis90/roddyduk $ lp lab10  
request id is hplaser-5 (1 file(s))
```

Or to the default printer

```
/home/cis90/roddyduk $ lpr lab10
```

CUPS

lp and lpr commands



```
/home/cis90/roddyduk $ echo "Print Me Loudly" | lp -d epson  
request id is epson-7 (1 file(s))
```

```
/home/cis90/roddyduk $ echo "Print Me Quietly" | lpr -P hplaser  
/home/cis90/roddyduk $
```

Note that both lp and lpr will read from stdin.

This allows output from another command to be piped in

Managing Print Jobs

CUPS

Rejecting Jobs



*Clicking the **Reject Jobs** button on the web based utility will reject further jobs*

```
[root@benji ~]# lp myfile  
lp: Destination "hp7550" is not accepting jobs.  
[root@benji ~]#
```

```
[root@benji ~]# lpr myfile  
lpr: Destination "hp7550" is not accepting jobs.  
[root@benji ~]#
```

CUPS

Stopping the Printer

```
[root@benji ~]# lp myfile
request id is hp7550-22 (1 file(s))
[root@benji ~]# lpr myfile
[root@benji ~]# lp myfile
request id is hp7550-24 (1 file(s))
[root@benji ~]# lpr myfile
```

```
[root@benji ~]# lpq
hp7550 is not ready
Rank   Owner   Job     File(s)
Total Size
1st    root    22     myfile
1024 bytes
2nd    root    23     myfile
1024 bytes
3rd    root    24     myfile
1024 bytes
4th    root    25     myfile
1024 bytes
```

```
[root@benji ~]# lpstat
hp7550-22          root          1024    Sat 15
Nov 2008 12:20:23 PM PST
hp7550-23          root          1024    Sat 15
Nov 2008 12:20:28 PM PST
hp7550-24          root          1024    Sat 15
Nov 2008 12:20:31 PM PST
hp7550-25          root          1024    Sat 15
Nov 2008 12:20:34 PM PST
```



*Clicking the **Stop Printer** button on the web based utility will still allow jobs to be spooled*

CUPS

Showing jobs waiting to print



```
[root@benji ~]# lpq
hp7550 is not ready
Rank   Owner   Job      File(s)
Total Size
1st    root    22      myfile
1024 bytes
2nd    root    23      myfile
1024 bytes
3rd    root    24      myfile
1024 bytes
4th    root    25      myfile
1024 bytes
```

*Use lpq or lpstat to show
spooled print jobs*

```
[root@benji ~]# lpstat
hp7550-22          root          1024    Sat 15
Nov 2008 12:20:23 PM PST
hp7550-23          root          1024    Sat 15
Nov 2008 12:20:28 PM PST
hp7550-24          root          1024    Sat 15
Nov 2008 12:20:31 PM PST
hp7550-25          root          1024    Sat 15
Nov 2008 12:20:34 PM PST
```

CUPS

Removing/canceling pending print jobs

```
[root@benji ~]# lpq
hp7550 is not ready
Rank   Owner   Job     File(s)
Total Size
1st    root    22      myfile
1024 bytes
2nd    root    23      myfile
1024 bytes
3rd    root    24      myfile
1024 bytes
4th    root    25      myfile
1024 bytes
```

```
[root@benji ~]# cancel 22
[root@benji ~]# cancel 23
[root@benji ~]# lprm 24
[root@benji ~]# lprm 25
```

```
[root@benji ~]# lpq
hp7550 is not ready
no entries
```

```
[root@benji ~]# lpstat
[root@benji ~]#
```

*Use lpq or lpstat to show
the spooled print jobs*

*Use cancel or lprm to
remove print jobs*





Wrap up

Commands:

lp, lpr	- Linux print command
cancel, lprm	- cancel print job
lpq, lpstat	- Show print queue

Web:

http://hostname:631	- CUPS web based management utility
http://hostname:9100	- HP JetDirect printer



Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

No Quiz

No Lab due

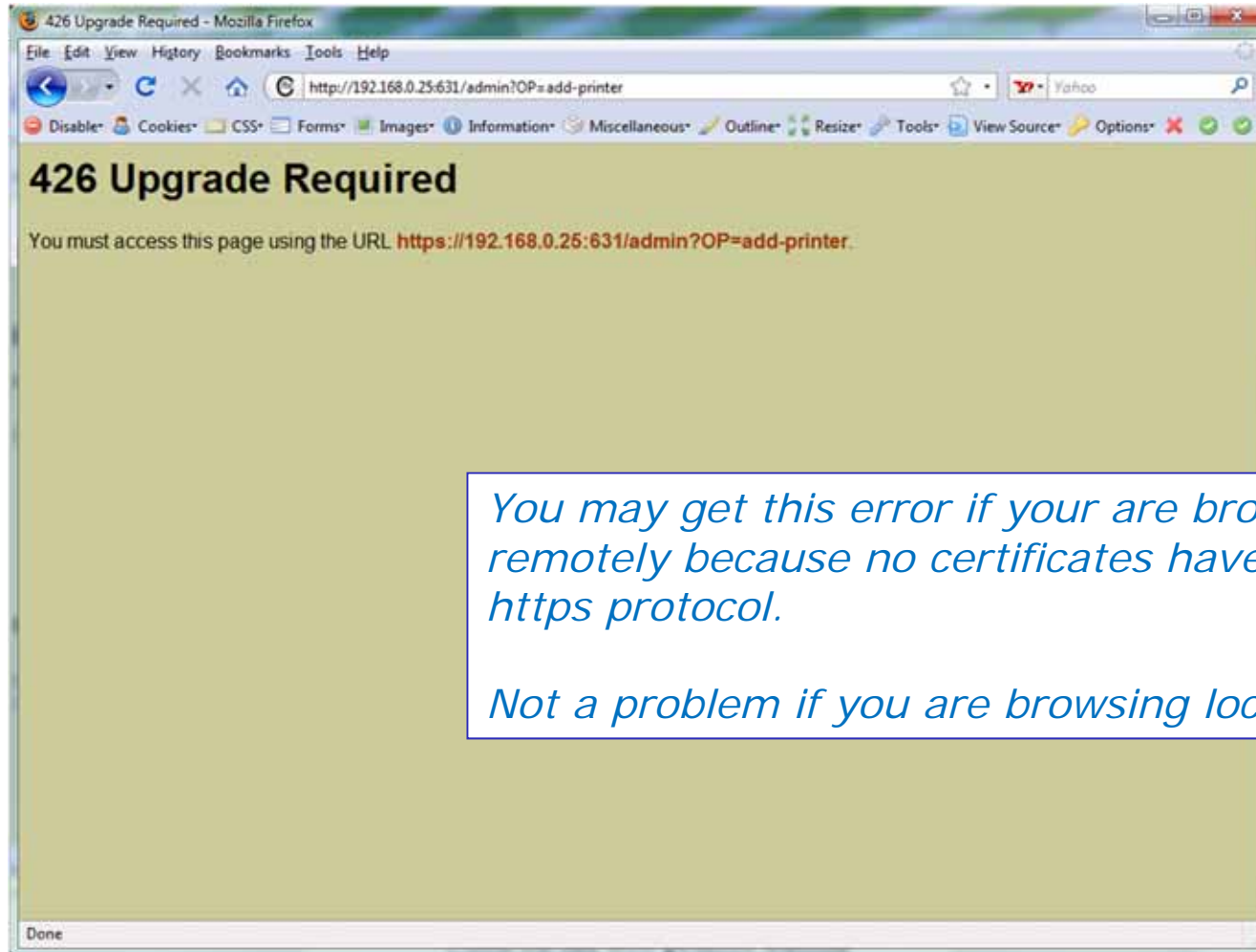
Work on final projects

Optional extra credit labs



Backup

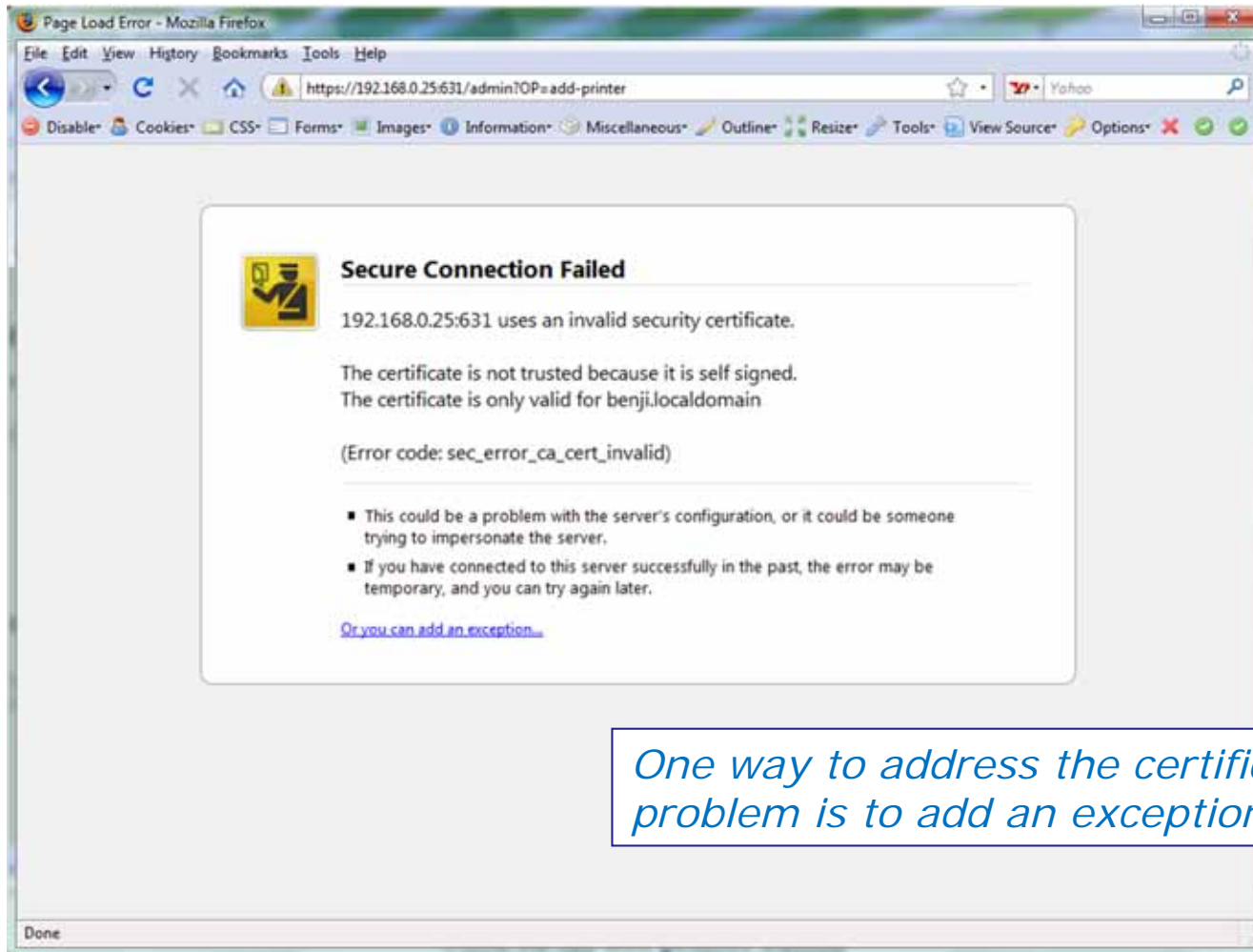
CUPS



You may get this error if your are browsing in remotely because no certificates have been set up for https protocol.

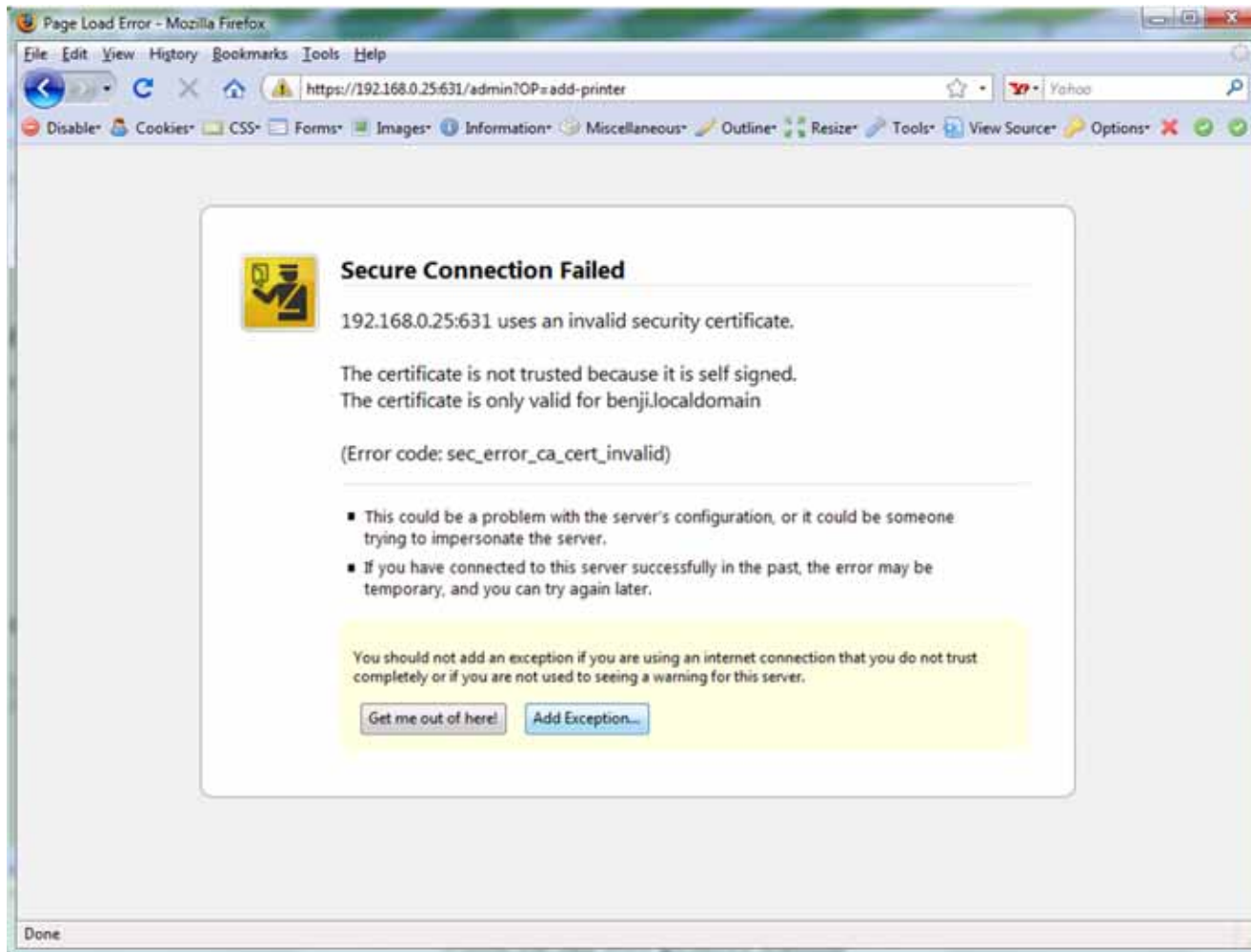
Not a problem if you are browsing locally

CUPS



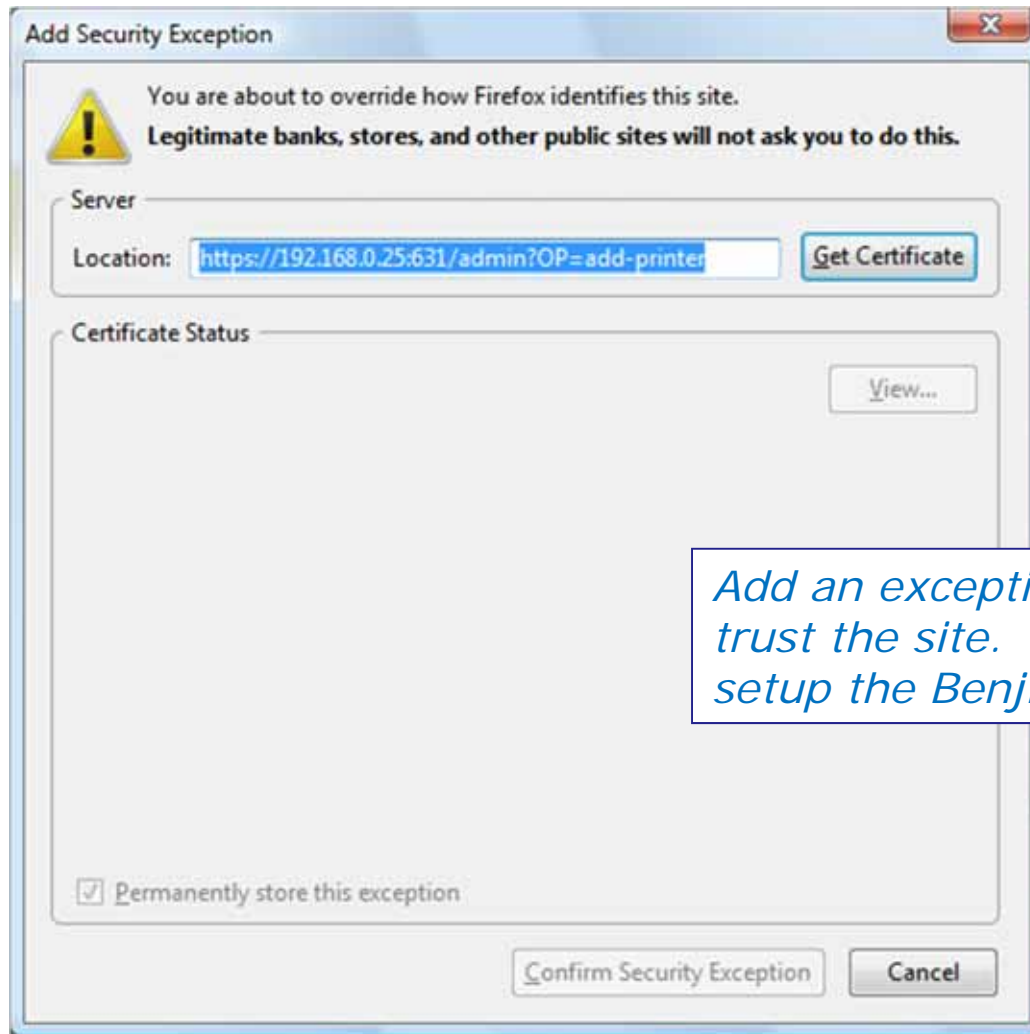
One way to address the certificate problem is to add an exception.

CUPS



This adds the exception

CUPS



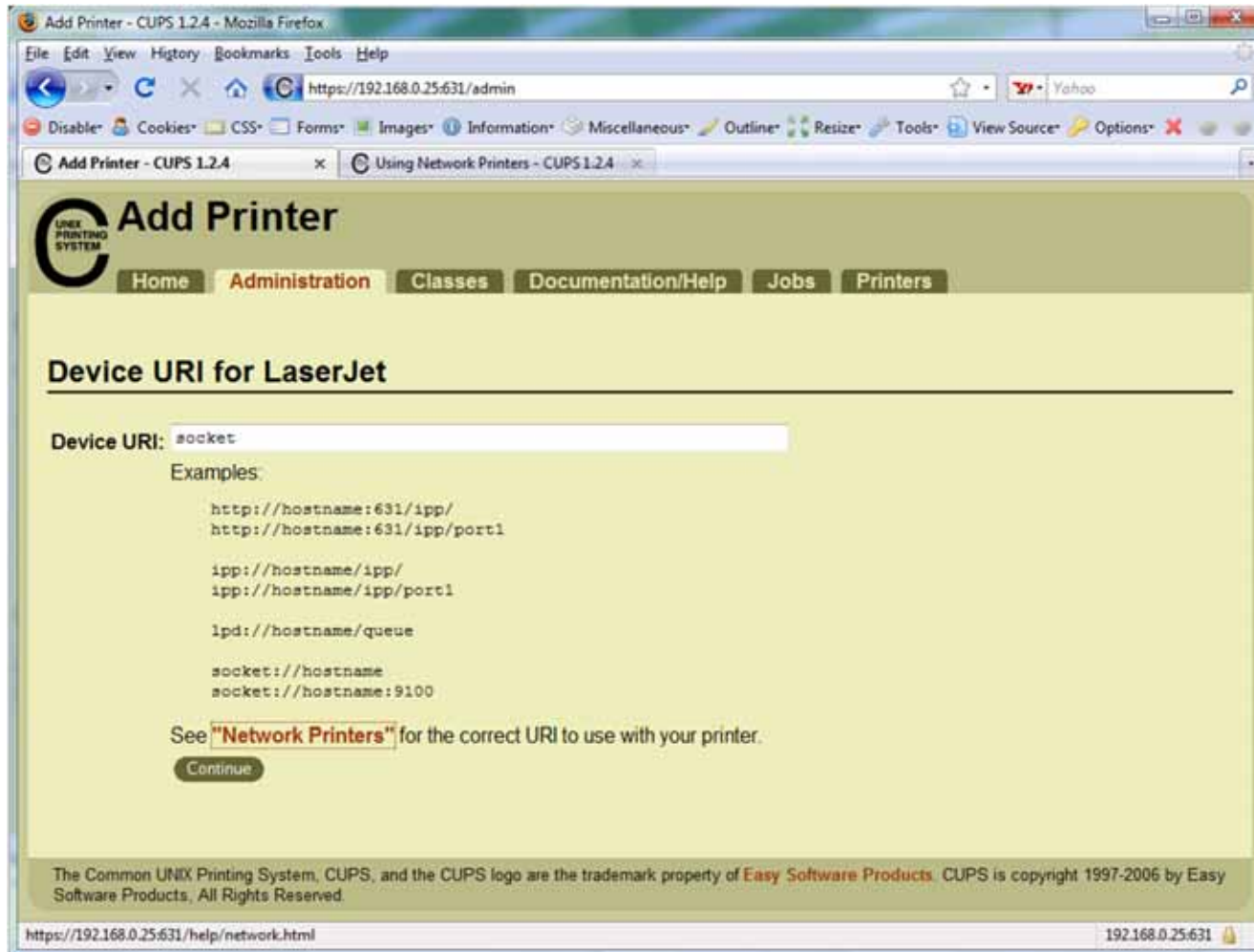
Add an exception only when you do trust the site. In this case we built and setup the Benji VM so we trust it.

CUPS



Click Confirm Security Exception button

CUPS



Hmmm lets click on Network Printers link to figure this one out.

CUPS

Using Network Printers - CUPS 1.2.4 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.0.25:631/help/network.html

Using Network Printers

Home Administration Classes Documentation/Help Jobs Printers

Search in Using Network Printers: Search Clear

Using Network Printers [View Printable Version](#)

Network Printer URIs

Once you have set the IP address you can access the printer or print server using the `ipp`, `lpd`, or `socket` backends. The following is a list of common network interfaces and printer servers and the settings you should use with CUPS.

Table 1: Common Device URIs

Model/Manufacturer	Device URI(s)
Apple LaserWriter	<code>lpd://address/PASSTHRU</code>
Axis w/o IPP	<code>socket://address:9100</code>
Axis OfficeBasic (see directions)	<code>socket://address:9101</code> <code>socket://address:9102</code>
Axis w/IPP	<code>ipp://address/LPT1</code> <code>ipp://address/LPT2</code> <code>ipp://address/COM1</code>

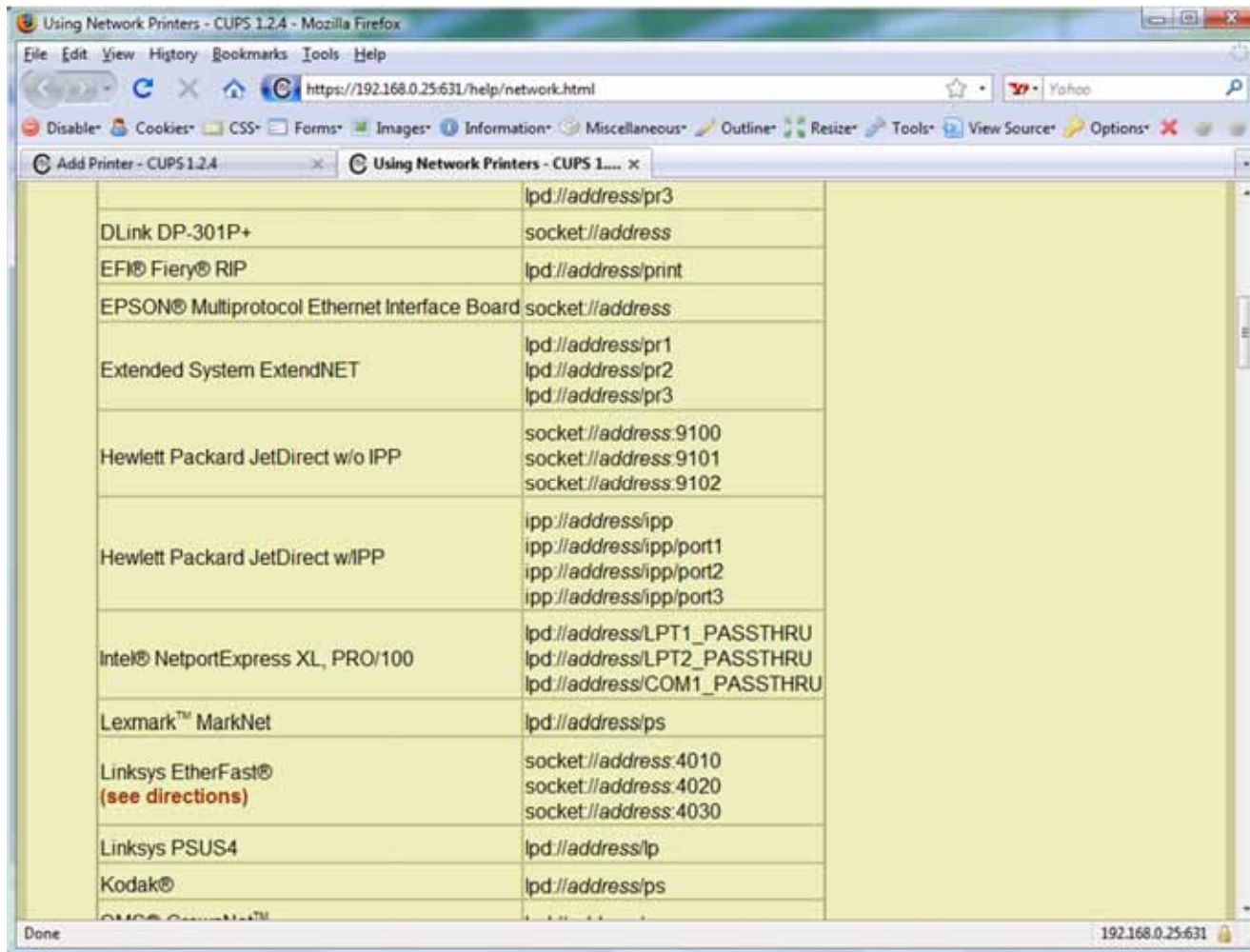
On-Line Help Documents

- All Documents
- Getting Started
- Man Pages
- Programming
- References
- Specifications

Done 192.168.0.25:631

*Scroll down to
HP printers*

CUPS



We will use the JetDirect w/o IPP for the HP 1320n

IPP is Internet Printing Protocol for send print jobs over the Internet via the http protocol

HP JetDirect cards use port 9100

A socket is the combination of an IP address and a port number.

CUPS

convert command

JPEG files need to be converted to postscript before printing with lp or lpr commands



```
[root@benji Desktop]# convert benji-500x420.jpg benji-500x420.ps
```

```
[root@benji Desktop]# lp benji-500x420.ps
```

```
request id is hp7550-29 (1 file(s))
```

```
[root@benji Desktop]# lpq
```

```
hp7550 is not ready
```

Rank	Owner	Job	File(s)	Total Size
1st	root	28	benji-500x420.ps	1284096 bytes
2nd	root	29	benji-500x420.ps	1284096 bytes

```
[root@benji Desktop]# cancel 29
```

```
[root@benji Desktop]# cd /var/spool/cups/
```

Print job #28

```
[root@benji cups]# ls
```

```
0000001b  c00009  c00012  c00015  c00018  c00021  c00024  c00027  d00028-001
c00001    c00010  c00013  c00016  c00019  c00022  c00025  c00028  tmp
c00008    c00011  c00014  c00017  c00020  c00023  c00026  c00029
```

```
[root@benji cups]# ls tmp
```

Process Information

Use `-l` for additional options

```
[rsimms@opus ~]$ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	201	6204	6203	0	75	0	-	1165	wait	pts/6	00:00:00	bash
0	R	201	6521	6204	0	77	0	-	1050	-	pts/6	00:00:00	ps

Annotations:

- Running or sleeping (points to the 'S' and 'R' status characters)
- User ID (points to the 'UID' column)
- Process ID (points to the 'PID' column)
- Parent Process ID (points to the 'PPID' column)
- Size in 1K blocks (points to the 'SZ' column)

Common Environment Variables

```

/home/cis90/roddyduk $ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/../bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=" "
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 002
set -o ignoreeof
stty susp
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `

/home/cis90/roddyduk $

```

On Opus, PS1 is set in /etc/bashrc and then redefined in .bash_profile

Common Environment Variables

```
/home/cis90/roddyduk $ cat .bashrc
```

```
# .bashrc
```

```
# User specific aliases and functions
```

```
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
```

```
    . /etc/bashrc
```

```
fi
```

```
alias print="echo -e"
```

```
/home/cis90/roddyduk $
```

```
/home/cis90/roddyduk $ cat /etc/bashrc | grep PS1
```

```
if [ "$PS1" ]; then
```

```
    [ "$PS1" = "\\s-\\v\\\\"$ " ] && PS1="[\u@\h \W]\\\\"$ "
```

```
/home/cis90/roddyduk $
```

On Opus, PS1 is set in /etc/bashrc and then redefined in .bash_profile 162

```
/home/cis90/roddyduk $ cat program
echo "program is being run"
fan=high
ac=on
export ac
alias copy=cp
```

*Use **vi** to make this file, then use **chmod** to give it execute permissions*

```
/home/cis90/roddyduk $ show
fan= ac=
-bash: type: copy: not found
```

Initial state

```
/home/cis90/roddyduk $ program
program is being run
/home/cis90/roddyduk $ show
fan= ac=
-bash: type: copy: not found
```

Not changed

```
/home/cis90/roddyduk $ source program
program is being run
/home/cis90/roddyduk $ show
fan=high ac=on
copy is aliased to `cp`
ac=on
```

Changed

Do you get the same results?

*Note: using **alias show='echo fan=\$fan ac=\$ac; type copy; env | grep ac'***