



## Lesson Module Checklist

- Slides -
- Properties -
- Flash cards -
- First minute quiz -
- Web calendar summary -
- Web book pages -
- Commands -
- Lab -
  
- CCC Confer wall paper -
- labx1 and project posted -
  
- Materials uploaded -
- Backup headset charged -
- Backup slides, CCC info, handouts on flash drive -
  
- Pickup Polycom
- Check that room headset is charged Aptos 831.479.6392 -



Dieskau



Jonathan



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



Ana



David



Obie



Dave



Cole



Corey



Nancy



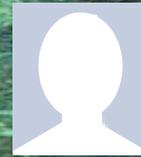
Ryan



Elia



Tasha



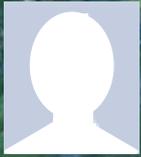
Darren



Scott



Devin



Everett



Juan



Raven



Rogan



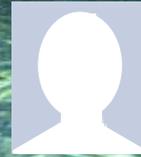
Mike



Mook



Melissa



Cameron



Jose



Jeff



Matt



Kenneth



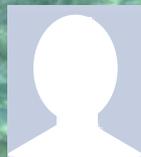
Ousmane



Ian



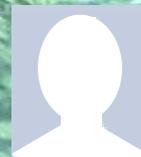
Solomon



Henry



Matthew



Mason



Chan

# The LAST Quiz

**email answers to: [risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)  
(within the first few minutes of class)**



- [ ] Has the phone bridge been added?
- [ ] Is recording on?
- [ ] Does the phone bridge have the mike?
- [ ] Share slides, putty x 3, and Chrome
- [ ] Disable spelling on PowerPoint

# Shell Scripting and Printing

## Objectives

- Be able to print, view the print queue and cancel print jobs

## Agenda

- Quiz
- Housekeeping
- Refresh
- Shell scripting
- Printing



# Questions

## Previous material and assignment

1. Previous material
2. Lab 10



# Life without a path

## The Path

- *The path is used by the shell to locate commands to run*
- *The path is defined by the contents of the PATH variable*
- *If the path is not defined then each command to run must be specified using it's absolute pathname*

```
/home/cis90/simben $ echo $PATH  
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:  
/home/cis90/bin:/home/cis90/simben/bin:.
```

## Class Activity: Life without a path

Remove your path variable:

```
/home/cis90/simben $ unset PATH
```

```
/home/cis90/simben $ echo $PATH
```

```
/home/cis90/simben $
```

*What just happened?*

## The Path

*If the path is not defined then each command to run must be specified using an absolute pathname*

```
/home/cis90/simben $ ls letter  
-bash: ls: No such file or directory
```



```
/home/cis90/simben $ /bin/ls letter  
letter  
/home/cis90/simben $
```

## The Path

*Some commands still work without a path ... why?*

```
/home/cis90/simben $ echo "I want my path back"  
I want my path back
```

```
/home/cis90/simben $ type echo  
echo is a shell builtin
```

```
/home/cis90/simben $ type type  
type is a shell builtin
```

## The Path

### Make a new path

```
/home/cis90/simben $ ls letter
-bash: ls: No such file or directory
```



*The **ls** command is in /bin so lets put that on the path*



```
/home/cis90/simben $ PATH=/bin
/home/cis90/simben $ ls letter
letter
```

```
/home/cis90/simben $ stat letter
-bash: stat: command not found
```



*The **stat** command is in /usr/bin so lets append that directory too*



```
/home/cis90/simben $ PATH=$PATH:/usr/bin
/home/cis90/simben $ stat letter
  File: `letter'
  Size: 1059          Blocks: 16          IO Block: 4096
regular file
Device: fd00h/64768d  Inode: 102594       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/simben90)  Gid: (
90/   cis90)
Access: 2012-04-30 15:43:28.000000000 -0700
Modify: 2012-03-20 10:31:30.000000000 -0700
Change: 2012-04-30 07:34:30.000000000 -0700
```

## The Path



```
/home/cis90/simben $ allscripts
-bash: allscripts: command not found
```



```
/home/cis90/simben $ PATH=$PATH:/home/cis90/bin
/home/cis90/simben $ allscripts
```

*The **allscripts** shell script is in /home/cis90/bin so lets add that directory to the path as well*

```
*****
*                               Spring 2012 CIS 90 Online Projects                               *
*****
1) Ana           10) Elia           19) Melissa      28) Solomon
2) Cameron      11) Henry           20) Mook         29) Tasha
3) Cole         12) Ian             21) Nancy
4) Corey        13) Jeffrey         22) Obie
5) Darrin       14) Jonathan       23) Ousmane
6) David H.    15) Juan            24) Raven
7) Dave R.     16) Mason           25) Rogan
8) Devin       17) Matthew A.     26) Ryan
9) Dieskau     18) Matt F.        27) Scott
```

```
*****
*                               Examples and Hall of Fame                               *
*****
50) Duke        51) Benji           52) Junious     53) Janet

99) Exit
```

Enter Your Choice:

## The Path

```
/home/cis90/simben $ datecal
bash: datecal: command not found
```



```
/home/cis90/simben $ ./bin/datecal
```

```
Tue May 8 14:28:43 PDT 2012
```

April 2012							May 2012							June 2012						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7			1	2	3	4	5						1	2
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23
29	30						27	28	29	30	31			24	25	26	27	28	29	30

*How can I run a script in my own bin directory without having to put a ./ in front of it?*

## The Path

*Easy ... add your own bin directory to the path*

```
/home/cis90/simben $ datecal  
bash: datecal: command not found
```



```
/home/cis90/simben $ PATH=$PATH:/home/cis90/simben/bin  
/home/cis90/simben $ datecal
```

```
Tue May 8 14:30:59 PDT 2012
```

April 2012							May 2012							June 2012							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
1	2	3	4	5	6	7			1	2	3	4	5							1	2
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9	
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16	
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23	
29	30						27	28	29	30	31			24	25	26	27	28	29	30	

```
/home/cis90/simben $
```

## The Path

```
/home/cis90/simben $ dogbone  
-bash: dogbone: command not found
```



```
/home/cis90/simben $ ./dogbone  
What is your name? Benji  
What is your favorite bone? Chicken  
Hi Benji, your favorite bone is Chicken
```

*How can I run a script in the current directory without having to put a ./ in front of it?*

## The Path

*Easy ... add "here" or "." to the path*

```
/home/cis90/simben $ dogbone  
-bash: dogbone: command not found
```



```
/home/cis90/simben $ PATH=$PATH:.  
/home/cis90/simben $ dogbone  
What is your name? Benji  
What is your favorite bone? Chicken  
Hi Benji, your favorite bone is Chicken
```

## The Path

### *Appending directories to the path*

```
/home/cis90/simben $ unset PATH  
/home/cis90/simben $ echo $PATH
```

```
/home/cis90/simben $ PATH=/bin  
/home/cis90/simben $ echo $PATH  
/bin
```

```
/home/cis90/simben $ PATH=$PATH:/usr/bin  
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin
```

```
/home/cis90/simben $ PATH=$PATH:/home/cis90/bin  
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin:/home/cis90/bin
```

```
/home/cis90/simben $ PATH=$PATH:/home/cis90/simben/bin  
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin:/home/cis90/bin:/home/cis90/simben/bin
```

```
/home/cis90/simben $ PATH=$PATH:..  
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin:/home/cis90/bin:/home/cis90/simben/bin:..
```

## The Path

### *Making the path permanent using .bash\_profile*

```

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

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

# User specific environment and startup programs

PATH=$PATH:/home/cis90/bin:$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/simben $

```



# Housekeeping

## Previous material and assignment

1. Lab 10 due midnight tonight
2. Extra Credit Labs X1 and X2 (30 points each)

Code	Grading	Quizzes & Tests															Forum			Lab						Project	Extra Credit	Total Grade				
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	T4	T5	F1	F2	F3	L1	L2	L3	L4	L5	L6				L7	L8	L9	L10
adaldrida	P/NP	3	1	3	2	3	3	3	2	3	28	20	20	20	20	20	20	20	24	26	30	27	30	29	30	28					19	
alatar	Grade	2	3	3	3	3	3	3	3	3	28	28	20	20	20	20	20	20	27	30	26	30	30	30	30					25		
amroth	Grade	3	3	3	3	3	3	3	3	3	28	12	20	20	20	20	20	20	29	23	26	29	27	28	20	27					20	
arador	Grade	3	3	3	3	3	3	3	3	3	25	21	20	20	20	20	20	19	30	24	29	30	30	30	30					29		
aragorn	Grade	3	3	3	3	3	3	3	3	3	28	30	20	20	20	20	20	20	28	15	28	26	28	30	30					33		
arwen	Grade	3	1	3	3	3	3	3	3	3	27	28	20	20	20	20	20	20	29	30	29	30	30	30	30					66		
celebrían	Grade	3	3	3	2	3	3	3	3	3	23	27	20	4	30				30	29	30	27	20	30					24			
cirdan	Grade	3	3	3	3	3	3	3	3	3	29	28	20	20	20	20	20	28	29	30	24	30	30	30	30					60		
dénéthor	Grade	3	2	3	3	3	3	3	3	3	20	0	0	0	14	22	28													7		
dori	Grade	3	2	3	2	3	3	3	3	3	23	23	16	20	12	28	27	30	30	20	30	20	29						17			
dwalin	Grade	3	1	1	1	1	3				19	22	20	20	20	20	20	28	25	28	22	18	29	20	27					25		
elrond	Grade	3	3	3	3	3	3	3	3	3	28	30	20	20	20	20	20	30	28	29	30	28	30	30					32			
eomer	Grade	2	2	1	2	3	3	3	3	3	18	10	20	20	20	20	20	21	25	30	23	23	28	25					12			
eowyn	Grade	2	3	3							26	20	20	20	8	26	28	30	15	30	28	20	30						25			
fródo	Grade	3	3	3	3	3	3	3	3	3	28	28	20	20	20	20	20	28	24	30	29	30	30	30					61			
gwaihir	Grade	1	3	2	3	3	3	3	3	3	22	12	20	20	20	20	20	25	22	30	18	26	19	23	20				16			
haldir	Grade	3	0	3	3	3	3	3	3	3	26	25	20	16	20	30	28	28	20	30	27	20	30						25			
isroth	Grade	2	3								12	8	8	0	0	21	30													5		
khamul	Grade	3	2	3	2	3	3	3	3	3	16	27	20	20	20	20	20	29	22	30	22	20	30	30					23			
nessa	Grade	1	3	3	3	3	3	1	0	0	24	23	20	20	20	20	30	24	28	20	18	30	30	26					14			
orome	Grade	3	3	3	3	3	3	3	3	3	27	24	20	16	16	30	27	30	25	30	30	30	30	30					19			
pippin	Grade	3	3	3	2	3	3	3	3	3	27	20	8	30	20	30	30	30												25		
quickbeam	Grade	3	3	3	3	3	3	3	3	3	27	28	20	20	20	20	25	29	30	28	29	26	30	30					23			
samwise	Grade	2	3	3	3						5	23	12	4	8	0	22	16	24	30										13		
shadowfax	Grade	2	3	3	3	3	3	3	3	3	27	28	20	20	20	20	20	14	29	30	30	30	30	30					29			
strider	Grade	3	3	3	3	3	3	3	3	3	28	27	20	20	20	20	30	28	30	29	30	30	30	30					60			
theoden	Grade	1	0	3	2	3	3	3	3	3	23	26	0	8	20	27	22	29	25	27	20	30							7			
treebeard	Grade	3	3	3	3	3	3	3	3	3	28	24	16	20	20	20	28	25	30	29	30	26	30	30					27			
tulkas	Grade	3	2	0	3	3	1	2	2	2	22	23	0	0	20	30	22	28	18	29	24	27	20	28					10			
ulmo	Grade	3	3	3	3	3	3	3	3	3	27	29	20	20	20	20	28	25	28	30	30	29	30	30					31			

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

*Be sure to monitor you own progress using the Grades page of the course website*

### Tally as of 5/6/2012

- adaldrida: 89% (374 of 417 points)
- alatar: 95% (397 of 417 points)
- amroth: 90% (379 of 417 points)
- arador: 91% (381 of 417 points)
- aragorn: 100% (418 of 417 points)
- arwen: 113% (474 of 417 points)
- celebrían: 72% (304 of 417 points)
- cirdan: 111% (465 of 417 points)
- dori: 88% (370 of 417 points)
- dwalin: 79% (332 of 417 points)
- elrond: 105% (439 of 417 points)
- eomer: 81% (340 of 417 points)
- eowyn: 86% (361 of 417 points)
- fródo: 111% (463 of 417 points)

- gwaihir: 82% (346 of 417 points)
- haldir: 87% (363 of 417 points)
- ioareth: 21% (90 of 417 points)
- khamul: 93% (389 of 417 points)
- nessa: 82% (342 of 417 points)
- orome: 99% (415 of 417 points)
- pippin: 47% (197 of 417 points)
- quickbeam: 101% (423 of 417 points)
- samwise: 39% (166 of 417 points)
- shadowfax: 103% (432 of 417 points)
- strider: 109% (456 of 417 points)
- theoden: 66% (276 of 417 points)
- treebeard: 100% (417 of 417 points)
- tulkas: 75% (315 of 417 points)
- ulmo: 96% (403 of 417 points)

*Remaining point earning opportunities = 3 + 20 + 30 + 60 = 113 points plus up to 90 points maximum extra credit*

13	5/9	<p><b>Quiz 10</b></p> <p><b>Printing and Shell Scripting</b></p> <ul style="list-style-type: none"> <li>. and exec commands</li> <li>UNIX printing</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>Presentation slides (<a href="#">download</a>)</li> </ul> <p><b>Assignment</b></p> <ul style="list-style-type: none"> <li><a href="#">Project</a></li> </ul> <p><b>Extra Credit Lab</b></p> <ul style="list-style-type: none"> <li><a href="#">Lab X1</a> (UNIX in Review)</li> <li><a href="#">Lab X2</a> (Pathnames)</li> </ul> <p><b>CCC Confer</b></p> <ul style="list-style-type: none"> <li><a href="#">Enter virtual classroom</a></li> <li><a href="#">Class archives</a></li> </ul>	9 (Gillay)	<a href="#">Lab 10</a>
14	5/16	<p><b>More Shell Scripting</b></p> <ul style="list-style-type: none"> <li>Examples</li> <li>Command categories</li> <li>scp command</li> <li>tar command</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>Presentation slides (<a href="#">download</a>)</li> </ul> <p><b>Assignment</b></p> <ul style="list-style-type: none"> <li><a href="#">Project</a></li> </ul> <p><b>CCC Confer</b></p> <ul style="list-style-type: none"> <li><a href="#">Enter virtual classroom</a></li> <li><a href="#">Class archives</a></li> </ul>	10 (Gillay)	
15	5/23	<p><b>File Transfer and Review</b></p> <ul style="list-style-type: none"> <li>Filezilla</li> <li>gzip and gunzip</li> <li>Project presentations</li> <li>Project workshop</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>Presentation slides (<a href="#">download</a>)</li> </ul> <p><b>Assignment</b></p> <ul style="list-style-type: none"> <li><a href="#">Prep for Test #3</a></li> </ul> <p><b>CCC Confer</b></p> <ul style="list-style-type: none"> <li><a href="#">Enter virtual classroom</a></li> <li><a href="#">Class archives</a></li> </ul>	10 (Gillay)	<a href="#">Project</a>
	5/30	<p><b>Test #3 (the final exam)</b></p> <p><b>Time</b></p> <ul style="list-style-type: none"> <li>1:00PM - 3:50PM in Room 2501</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>Presentation slides (<a href="#">download</a>)</li> <li>Test (<a href="#">download</a>)</li> </ul>		<a href="#">5 posts</a> <a href="#">Lab X1</a> <a href="#">Lab X2</a>

*Use the Calendar page to plan your endgame!*

*Project due*

*Final Exam (in room only, no CCC Confer)  
Extra Credit Labs Due  
Five forum posts*

## Fall 2012 Linux Classes

<input type="checkbox"/>	Fall 2012	Open	<a href="#">CIS-90-76505 (76505) Intro to UNIX/Linux</a>	Online Course	08/29/2012-12/12/2012 DE Internet Delay Inter, CVC Wednesday 01:15PM - 04:20PM, Online Course, Room OL	R. Simms	21 / 24 / 0	3.00	
<input type="checkbox"/>	Fall 2012	Open	<a href="#">CIS-98-77651 (77651) UNIX/Linux Shell Programming</a>	Online Course	08/27/2012-12/15/2012 DE Internet Delay Inter, CVC Tuesday 01:00PM - 04:05PM, Online Course, Room OL (more)...	J. Griffin	23 / 24 / 0	4.00	

### CIS 90 Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools. Recommended Preparation: CS 1L or CIS 172.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
76505	W	01:15PM-04:20PM	3.00	R.Simms	OL

Section 76505 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

### CIS 98 UNIX/Linux Shell Programming

Presents an introduction to shell programming in a UNIX/Linux environment, and is designed for system administrators or technical users with little or no programming background. Prerequisite: CIS 90. Repeatability: May be take 2 times.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
77651	T	01:00PM-04:05PM	4.00	J.Griffin	OL
&	Arr.	Arr.		J.Griffin	OL

Section 77651 is an ONLINE course. Required orientation Tuesday, 8/28 from 1:00PM-4:05PM in Room 2501. Meets weekly throughout the semester online during the scheduled times. Students attend online using CCC Confer. Additional 5 hr 5 min online lab per week. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

## Use the forum effectively to get scripting help

*Not so good ...*

**Preview:**

Help!

My script is getting weird error

- Homer

*Not enough information has been provided  
on this post for others to help*

## Use the forum effectively to get scripting help

*Better ... but requires viewer to log into Opus and you may have modified the script since posting*

### Preview:

Help!

My script is getting weird error

My script is here:

/home/cis90/milhom/bin/myscript

And this is the error:

CODE: SELECT ALL

```
/home/cis90/simben/bin $ ./script99
simben90
-rwxr-x--- 1 simben90 cis90 10489 Apr 30 07:33 /home/cis90/simben/bin/myscript
./script99: line 8: unexpected EOF while looking for matching `"'
./script99: line 16: syntax error: unexpected end of file
/home/cis90/simben/bin $
```

- Homer

*This post provides the location of the script and the error message which enables others to help you find and fix the problem*

## Use the forum effectively to get scripting help



### Preview:

Help!

My script is getting weird error

This is the script:

CODE: SELECT ALL

```
#!/bin/bash
# Test script
#
echo $LOGNAME
dir=/home/cis90/simben
ls -l $dir/bin/myscript
if [ -f "$dir/bin/myscript" ]; then
    echo you have a myscript file in the bin directory
else
    echo there is no myscript file in your bin directory!
fi
exit
```

And this is the error:

CODE: SELECT ALL

```
/home/cis90/simben/bin $ ./script99
simben90
-rwxr-x--- 1 simben90 cis90 10489 Apr 30 07:33 /home/cis90/simben/bin/myscript
./script99: line 8: unexpected EOF while looking for matching `"'
./script99: line 16: syntax error: unexpected end of file
/home/cis90/simben/bin $
```

- Homer

*Best ...*

*This post shows both the script and the error using code tags which enables others to help you find and fix the problem.*

*The thread will also benefit future CIS 90 students*



# The echo command

# 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 use the echo command** to provide for interaction and feedback 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
```



# final project permissions

# Final Project

```
rsimms@opus:/home/cis90/bin

*****
*           Spring 2012 CIS 90 Online Projects           *
*****
1) Ana          10) Elia          19) Melissa       28) Solomon
2) Cameron     11) Henry          20) Mook         29) Tasha
3) Cole        12) Ian             21) Nancy
4) Corey       13) Jeffrey         22) Obie
5) Darrin      14) Jonathan        23) Ousmane
6) David H.   15) Juan            24) Raven
7) Dave R.    16) Mason           25) Rogan
8) Devin      17) Matthew A.     26) Ryan
9) Dieskau    18) Matt F.        27) Scott

*****
*           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***

# Permissions

*A past forum post ...*

**Ha Ha Class**  
Dby on Tue May 12, 2009 12:22 pm

I'm sure this is some kind of payback for last week "Hacking" attempt 😄



```
File Edit View Terminal Help
~/bin/boom
# menu: A simple menu example
while true
do
clear
echo -n "
~~~~~ will fail his Final Project
1) Job 1
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Command for Task 1
echo "~~~~~ got hacked!!!!"
echo "what is your name?"
read NAME
echo "what are ur hobbies?"
"myscript" 42L, 646C
```

I will find out who did this 😄😄

~~~~~

ps. Im going to pass 😄

*Uh, oh ... someone got hacked!*

# Permissions

```
rsimms@opus:/home/cis90
[rsimms@opus cis90]$ ls -l /home/cis90/*/bin/myscript
-rwxrwxr-x 1 ahrmat90 cis90 536 May 2 15:10 /home/cis90/ahrmat/bin/myscript
-rwxr-x--- 1 bodian90 cis90 629 May 2 15:08 /home/cis90/bodian/bin/myscript
-rw-rw-r-- 1 deltas90 cis90 482 May 2 14:39 /home/cis90/deltas/bin/myscript
-rwxrwxr-x 1 doucor90 cis90 481 May 2 15:36 /home/cis90/doucor/bin/myscript
-rwxr-xr-x 1 flamat90 cis90 952 May 2 15:22 /home/cis90/flamat/bin/myscript
-rwxrwxr-x 1 lowmic90 cis90 509 May 2 15:01 /home/cis90/lowmic/bin/myscript
-rwxr-xr-x 1 macrya90 cis90 481 May 2 14:43 /home/cis90/macrya/bin/myscript
-rwxrwxr-x 1 maxsco90 cis90 613 May 2 15:13 /home/cis90/maxsco/bin/myscript
-rwxrwxr-x 1 milhen90 cis90 481 May 2 14:43 /home/cis90/milhen/bin/myscript
-rwxrwxr-x 1 pacnan90 cis90 671 May 2 15:14 /home/cis90/pacnan/bin/myscript
-rwxrwxr-x 1 porjon90 cis90 599 May 2 15:49 /home/cis90/porjon/bin/myscript
-rw-rw-r-- 1 pummas90 cis90 263 May 2 16:04 /home/cis90/pummas/bin/myscript
-rwxrwxr-x 1 reedie90 cis90 651 May 2 15:20 /home/cis90/reedie/bin/myscript
-rwxr-x--- 1 rodduk90 cis90 3198 Apr 30 07:49 /home/cis90/rodduk/bin/myscript
-rwxrw-r-- 1 shidev90 cis90 909 May 2 15:37 /home/cis90/shidev/bin/myscript
-rwxr-x--- 1 simben90 cis90 10489 Apr 30 07:33 /home/cis90/simben/bin/myscript
-rwxrwxr-x 1 varana90 cis90 627 May 2 15:12 /home/cis90/varana/bin/myscript
-rwxrwxr-x 1 veleli90 cis90 669 May 2 15:28 /home/cis90/veleli/bin/myscript
[rsimms@opus cis90]$
```

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

### *After Lab 10*

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

*Because your umask setting gives group members 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/cis90/simben $ vi mydate  
/home/cis90/simben $ cat mydate
```

```
_____  
/home/cis90/simben $ chmod +x mydate  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```



# 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 `/usr/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.*

```
function runningScript ()  
{
```

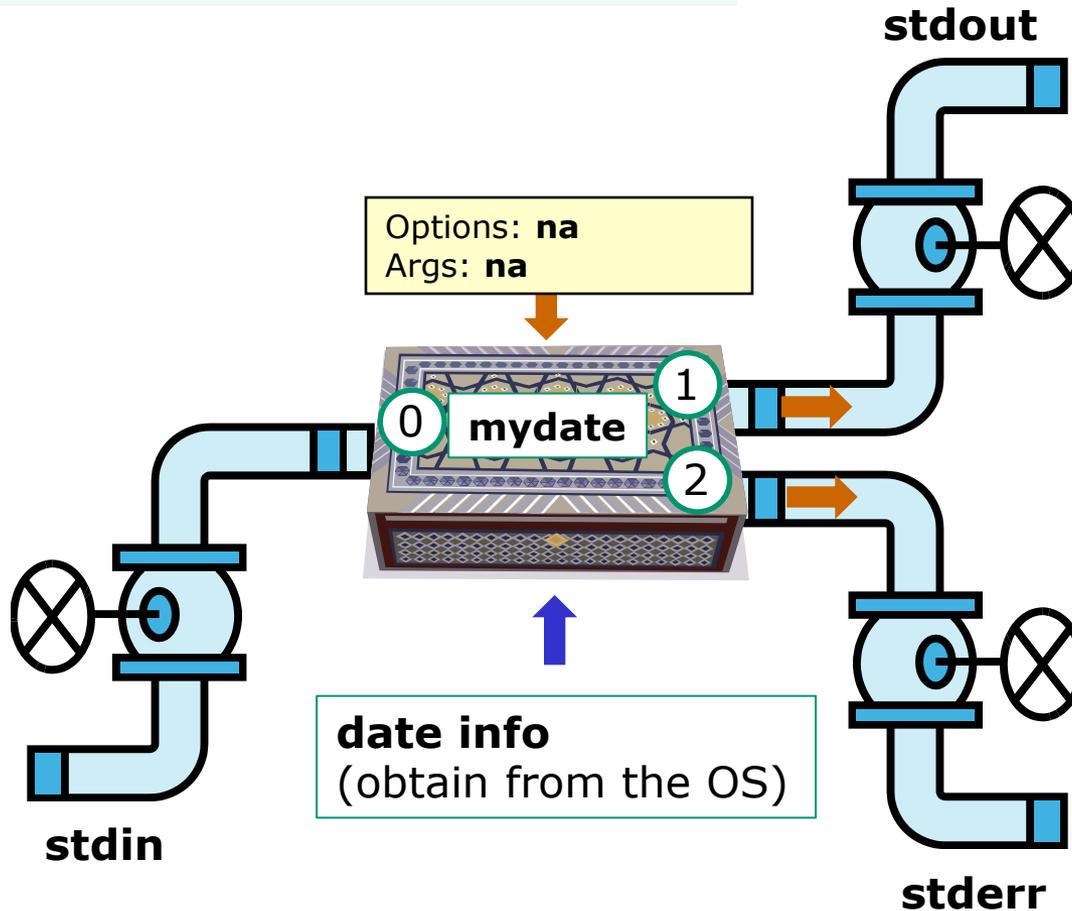
## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y'  
echo $myvar1 $myvar2 $myvar3
```

*Add this line to  
the last script we  
made*

# Running a Script

```
$ mydate
```



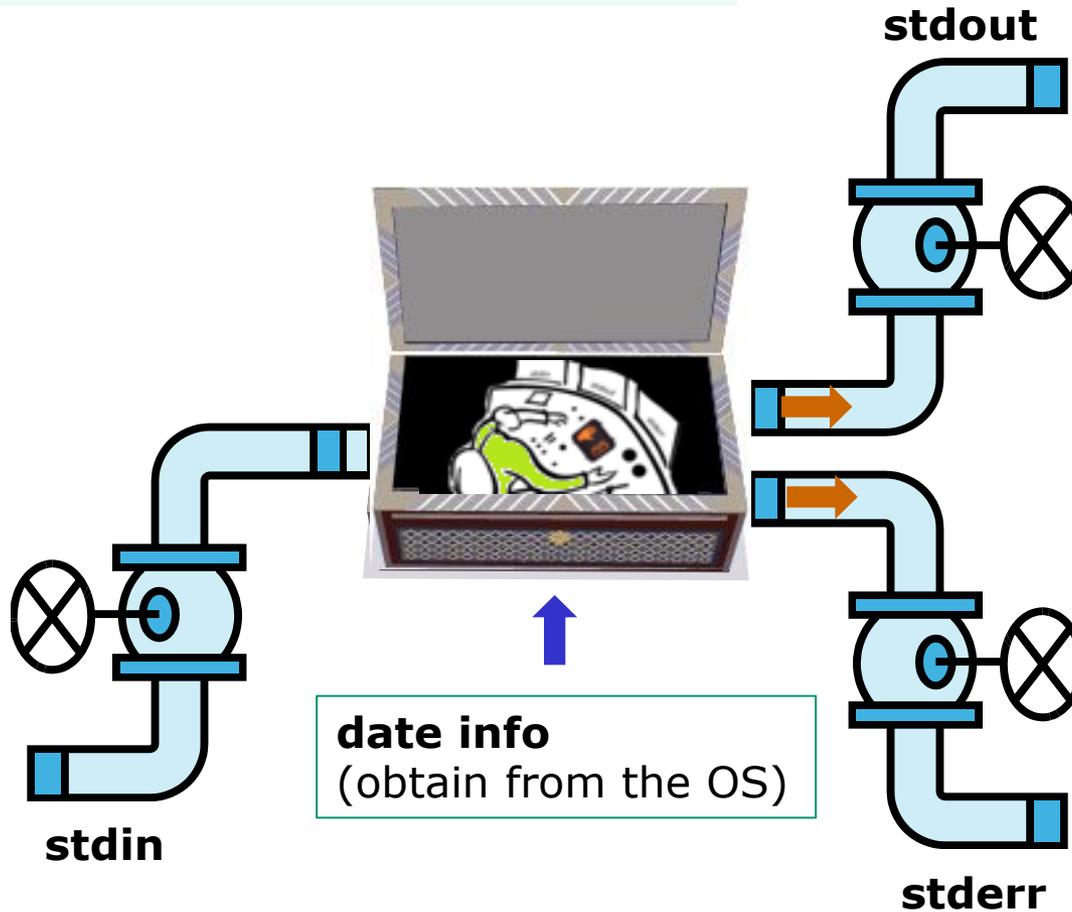
```
Hola simben90  
05/09/2012
```

*In this example, output from **myscript** goes to **stdout**.*

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

# Running a Script

```
$ mydate
```

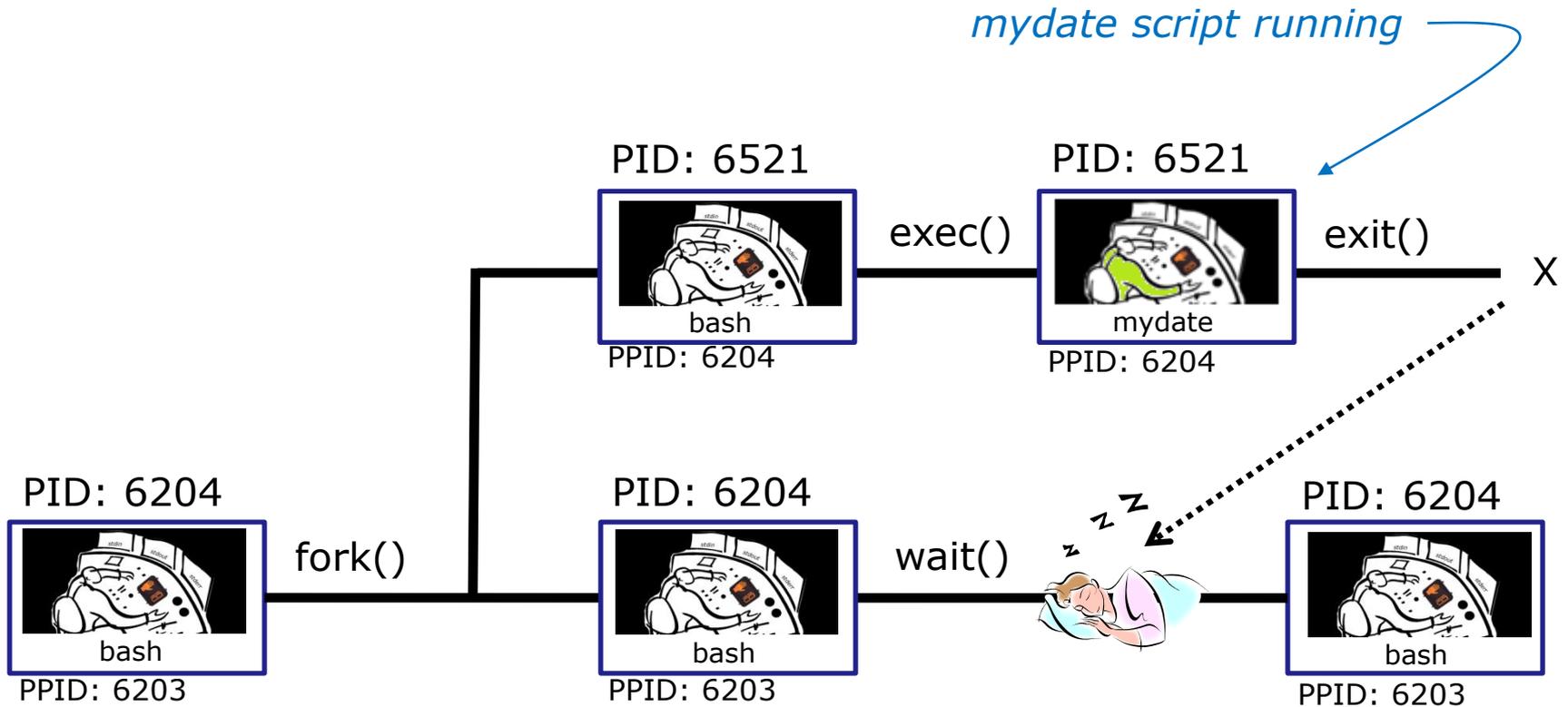


```
Hola simben90  
05/09/2012
```

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



# Running a Script



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

## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y'  
echo $myvar1 $myvar2 $myvar3
```

*In the parent process, set the three variables*

```
/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe  
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

*What happens if we run **mydate** now?*

## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y'  
echo $myvar1 $myvar2 $myvar3
```

```
/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe  
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```

*Running **mydate**  
(as a child process)*

```
/home/cis90/simben $
```

*Why no Tic Tac Toe output?*

## Running a Script

```
/home/cis90/simben $ export myvar1  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```

Tic

```
/home/cis90/simben $ export myvar2  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```

Tic Tac

```
/home/cis90/simben $ export myvar3  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```

Tic Tac Toe

*Rule 1: Children only  
see the variables you  
export*

## Running a Script

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ cat mydate
```

```
#!/bin/bash
```

```
echo "Hola $LOGNAME"
```

```
date +%m/%d/%Y'
```

```
echo $myvar1 $myvar2 $myvar3
```

```
myvar1=red myvar2=white myvar3=blue
```

```
echo $myvar1 $myvar2 $myvar3
```

*Add these  
new lines*

```
/home/cis90/simben $ mydate
```

```
Hola simben90
```

```
05/09/2012
```

```
Tic Tac Toe
```

```
red white blue
```

*Rule 2: children cannot  
make changes to the  
parent's variables*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
```

```
Tic Tac Toe
```

## Running a Script

*Unless we want them to*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ source mydate  
Hola simben90  
05/09/2012  
Tic Tac Toe  
red white blue
```

*Sourcing a script causes the instructions to be run in the parent process. A child process is not created*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
red white blue
```



```
}  
while no-comprende  
do  
  runningScript  
done
```



# 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

# CUPS

*Next step is to add printers*



*Printer: HP LaserJet 1320n  
Connection: LAN*

# CUPS

*The LaserJets also have a web-based management utility*



*IP Address for this 1320n  
is 172.30.1.14*

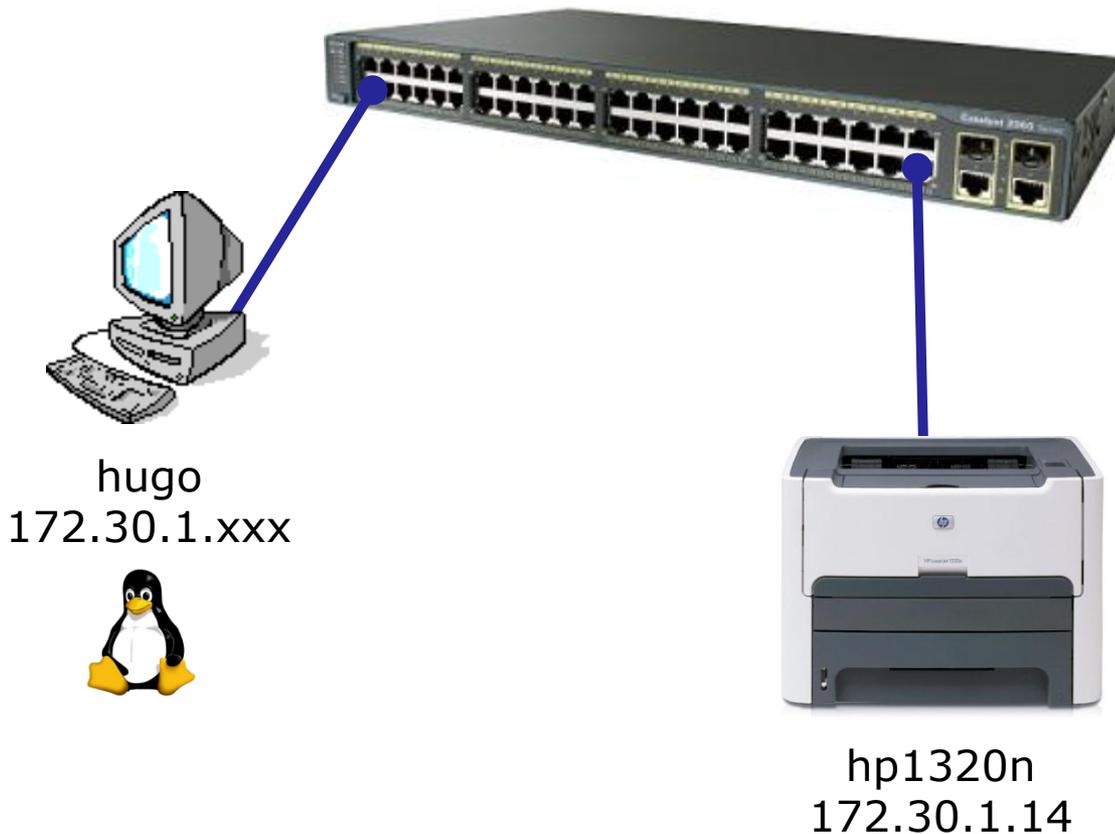
The screenshot shows a web browser window displaying the HP LaserJet 1320 series management utility. The browser address bar shows the IP address 172.30.1.14. The page has a blue header with the HP logo and the text "hp LaserJet 1320 series". Below the header, there are tabs for "Information", "Settings", and "Networking". The "Information" tab is selected, showing a "Device Status" section with a "Status: Ready" indicator and buttons for "Refresh Status", "Enter", and "Cancel Job". Below this is a "Supplies" section showing "Toner: (% Remaining)" and "Black Cartridge 97%" with a progress bar. At the bottom, there is a "Product Information" section with a table of device details.

|                        |                         |
|------------------------|-------------------------|
| Product Name:          | hp LaserJet 1320 series |
| Formatter Number:      | JH03T2Z                 |
| Product Serial Number: | CNHC6360LV              |
| Service ID:            | 16101                   |
| Firmware Datecode:     | 20041024                |
| Total Memory:          | 16 MBytes               |

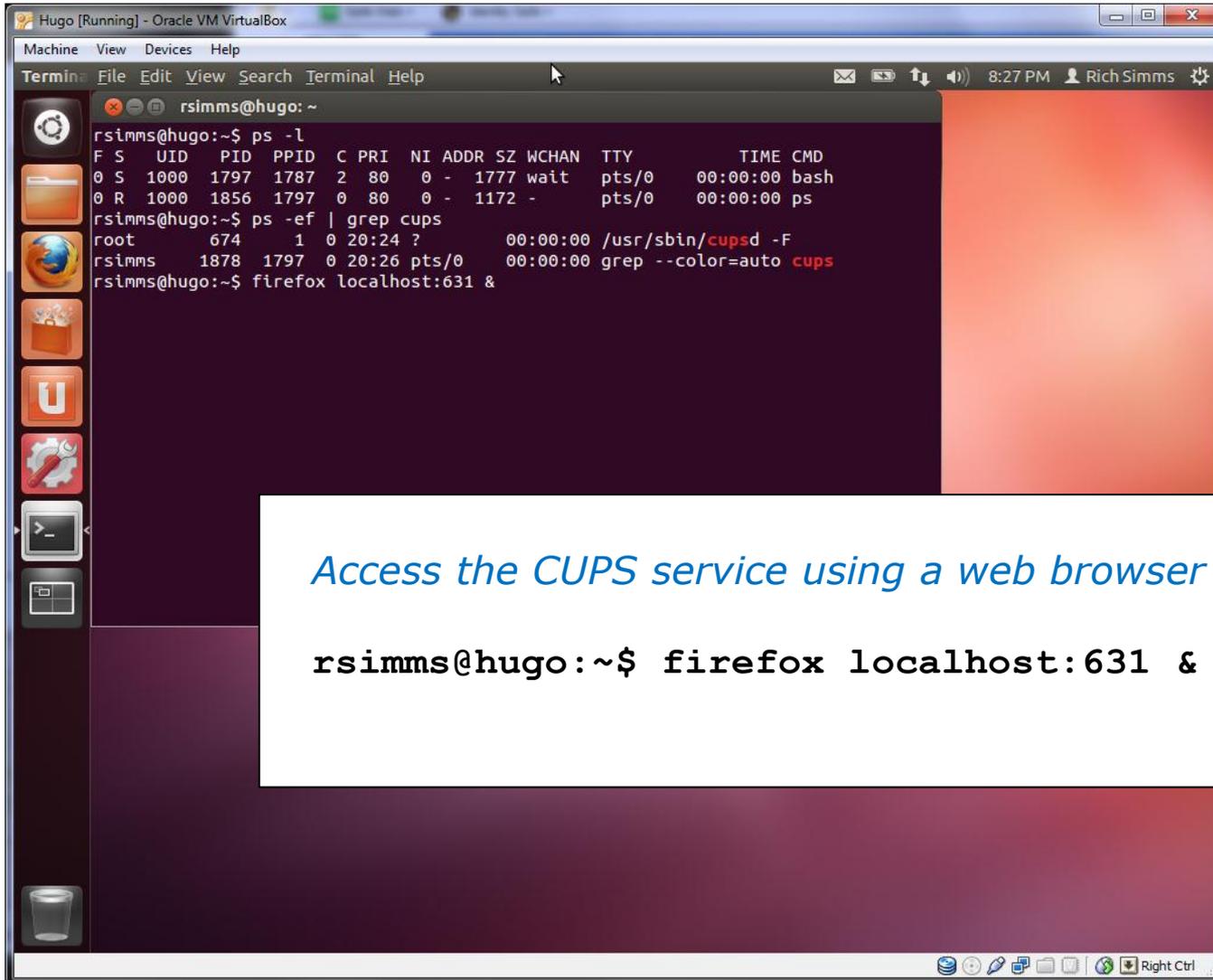
# Newest CUPS

# CUPS

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



# CUPS



Hugo [Running] - Oracle VM VirtualBox

Machine View Devices Help

File Edit View History Bookmarks Tools Help

Home - CUPS 1.5.2

localhost:631

Google

Home Administration Classes Online Help Jobs Printers Search Help

## CUPS 1.5.2

CUPS is the standards-based, open source printing system developed by [Apple Inc.](#) for Mac OS® X and other UNIX®-like operating systems.



### CUPS for Users

- [Overview of CUPS](#)
- [Command-Line Printing and Options](#)
- [What's New in CUPS 1.5](#)
- [User Forum](#)

### CUPS for Administrators

- [Adding Printers and Classes](#)
- [Managing Operation Policies](#)
- [Printer Accounting Basics](#)
- [Server Security](#)
- [Using Kerberos Authentication](#)
- [Using Network Printers](#)
- [cupsd.conf Reference](#)
- [Find Printer Drivers](#)

### CUPS for Developers

- [Introduction to CUPS Programming](#)
- [CUPS API](#)
- [Filter and Backend Programming](#)
- [HTTP and IPP APIs](#)
- [PPD API](#)
- [Raster API](#)
- [PPD Compiler Driver Information File Reference](#)
- [Developer Forum](#)

CUPS and the CUPS logo are trademarks of [Apple Inc.](#) CUPS is copyright 2007-2012 Apple Inc. All rights reserved.

The screenshot shows a web browser window with the address bar displaying '172.30.1.101:631'. The page title is 'Home - CUPS 1.5.2'. The browser's address bar shows '172.30.1.101:631'. The page has a navigation menu with 'Home', 'Administration', 'Classes', 'Online Help', 'Jobs', and 'Printers'. A search box labeled 'Search Help' is also present. The main content area features the heading 'CUPS 1.5.2' and a description: 'CUPS is the standards-based, open source printing system developed by Apple Inc. for Mac OS® X and other UNIX®-like operating systems.' To the right is the 'UNIX PRINTING SYSTEM' logo. Below this, there are three columns of links: 'CUPS for Users' (Overview of CUPS, Command-Line Printing and Options, What's New in CUPS 1.5, User Forum), 'CUPS for Administrators' (Adding Printers and Classes, Managing Operation Policies, Printer Accounting Basics, Server Security, Using Kerberos Authentication, Using Network Printers, cupsd.conf Reference, Find Printer Drivers), and 'CUPS for Developers' (Introduction to CUPS Programming, CUPS API, Filter and Backend Programming, HTTP and IPP APIs, PPD API, Raster API, PPD Compiler Driver Information File Reference, Developer Forum). A footer note states 'CUPS and the CUPS logo are trademarks of Apple Inc., registered in the U.S. and other countries. All other trademarks are the property of their respective owners.'

*Access the CUPS service remotely using a web browser on a different system*

The screenshot shows a web browser window titled "Administration - CUPS 1.5.2" with the address bar at "172.30.1.101:631/admin". The navigation menu includes "Home", "Administration" (which is highlighted), "Classes", "Online Help", "Jobs", and "Printers". There is also a "Search Help" input field. The main content area is divided into several sections: "Printers" with buttons for "Add Printer", "Find New Printers", and "Manage Printers"; "Classes" with "Add Class" and "Manage Classes"; "Jobs" with "Manage Jobs"; "Server" with "Edit Configuration File", "View Access Log", "View Error Log", and "View Page Log"; "Server Settings:" with an "Advanced" expandable section containing several checkboxes: "Show printers shared by other systems", "Share printers connected to this system" (with a sub-option "Allow printing from the Internet"), "Allow remote administration" (checked), "Use Kerberos authentication (FAQ)", "Allow users to cancel any job (not just their own)", and "Save debugging information for troubleshooting"; and "RSS Subscriptions" with an "Add RSS Subscription" button. Below the "Add RSS Subscription" button is a form with a "Name" field containing a slash "/" and a "Cancel RSS Subscription" button. At the bottom left, there is a small note: "CUPS and the CUPS logo are trademarks of".

Select the Administration tab to add printers

The screenshot shows a web browser window titled "Administration - CUPS 1.5.2" with the URL "https://172.30.1.101:631/admin/". The browser's address bar shows a red "X" over the URL. The page has a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers". The main content area is divided into sections: "Printers" (with buttons for "Add Printer", "Find New Printers", "Manage Printers"), "Classes" (with "Add Class" and "Manage Classes"), "Jobs" (with "Manage Jobs"), and "RSS Subscriptions" (with "Add RSS Subscription"). A "Server Settings" section is partially visible. An "Authentication Required" dialog box is overlaid on the page, containing the text: "The server 172.30.1.101:631 requires a username and password. The server says: CUPS." Below this text are input fields for "User Name" (containing "rsimms") and "Password" (containing "\*\*\*\*\*"). At the bottom of the dialog are "Log In" and "Cancel" buttons.

*Must authenticate to add new printer*

**Add Printer**

**Local Printers:**

- 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)

**Other Network Printers:**

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

*Nice! CUPS service already discovered a printer on the network*

**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://172.30.1.14

**Sharing:**  Share This Printer

*Customize printer description*

**Add Printer**

**Name:** HP\_LaserJet\_1320\_series  
**Description:** HP LaserJet 1320 series  
**Location:** Family room  
**Connection:** socket://172.30.1.14  
**Sharing:** Do Not Share This Printer  
**Make:** HP   
**Model:** HP LaserJet 1320 Series hpjls pcl3, 3.12.2 (en)  
HP LaserJet 1320 Series pcl3, hpcups 3.12.2 (en)  
HP 910 hpjls, 3.12.2 (en)  
HP 910, hpcups 3.12.2 (en)  
HP 915 hpjls, 3.12.2 (en)  
HP 915, hpcups 3.12.2 (en)  
HP 2000C Foomatic/pcl3 (en)  
HP 2000c hpjls, 3.12.2 (en)  
HP 2000c, hpcups 3.12.2 (en)  
HP 2500C Foomatic/pcl3 (en)

**Or Provide a PPD File:**  No file chosen

*Select the printer driver*

The screenshot shows a web browser window titled "Set Printer Options - CUPS". The address bar shows the URL "https://172.30.1.101:631/admin". The browser interface includes a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers", along with a "Search Help" field. The main content area is titled "Set Default Options for HP\_LaserJet\_1320\_series". Below the title is a button labeled "Query Printer for Default Options". A horizontal menu contains "General", "Printout Mode", "Banners", and "Policies", with "General" selected. Under the "General" section, there are four dropdown menus: "Media Size" (set to "Letter"), "Printout Mode" (set to "Normal"), "Media Source" (set to "Printer default"), and "Double-Sided Printing" (set to "Off"). At the bottom of the form is a button labeled "Set Default Options".

*Set default printing options for new printer*

HP\_LaserJet\_1320\_series - C x

https://172.30.1.101:631/printers/HP\_LaserJet\_1320\_series

Safe Web Identity Safe

Home Administration Classes Online Help Jobs Printers Search Help

### HP\_LaserJet\_1320\_series (Idle, Accepting Jobs, Not Shared)

Maintenance Administration

**Description:** HP LaserJet 1320 series  
**Location:** Family room  
**Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing)  
**Connection:** socket://172.30.1.14  
**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.

*Ready to roll!*

The screenshot shows a web browser window with the URL `https://172.30.1.101:631/printers/HP_LaserJet_1320_series`. The browser's address bar and tabs are visible. The page content includes a navigation menu with 'Printers' selected, a search bar, and a main heading: **HP\_LaserJet\_1320\_series (Processing, Accepting Jobs, Not Shared)**. Below the heading are dropdown menus for 'Maintenance' and 'Administration'. The printer's description, location, driver, connection, and defaults are listed. A 'Jobs' section contains a search bar and buttons for 'Show Completed Jobs' and 'Show All Jobs'. A table shows one active job with columns for ID, Name, User, Size, Pages, State, and Control. A text box at the bottom of the screenshot contains the text *Printing a test page*.

**HP\_LaserJet\_1320\_series (Processing, Accepting Jobs, Not Shared)**

Maintenance Administration

**Description:** HP LaserJet 1320 series  
**Location:** Family room  
**Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing)  
**Connection:** socket://172.30.1.14  
**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

Showing 1 of 1 active job.

| ID                        | Name    | User     | Size | Pages   | State            | Control             |
|---------------------------|---------|----------|------|---------|------------------|---------------------|
| HP_LaserJet_1320_series-1 | Unknown | Withheld | 1k   | Unknown | processing since | Cancel Job Move Job |

*Printing a test page*

The screenshot shows a web browser window with the URL `https://172.30.1.101:631/printers/HP_LaserJet_1320_series`. The page title is **HP\_LaserJet\_1320\_series (Idle, Accepting Jobs, Not Shared)**. Below the title, there are two dropdown menus for 'Maintenance' and 'Administration'. The main content area lists printer details: **Description:** HP LaserJet 1320 series, **Location:** Family room, **Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing), **Connection:** socket://172.30.1.14, and **Defaults:** job-sheets=none, none media=na\_letter\_8.5x11in sides=one-sided. A 'Jobs' section contains a search bar with the text 'Search in HP\_LaserJet\_1320\_series:' and buttons for 'Search' and 'Clear'. Below the search bar are buttons for 'Show Completed Jobs' and 'Show All Jobs'. The text 'No jobs.' is displayed in the center of the jobs section. A blue text box at the bottom of the screenshot contains the text *No print jobs currently spooled*.

# Printing in Linux

# Printing Commands

## **ATT System V based print subsystem**

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

## **BSD (Berkeley Software Distribution) 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

*BSD is a branch of UNIX developed at the University of California, Berkeley*

# CUPS

## lpstat command



Use **lpstat** to show spooled print jobs, available and default printers

```
rsimms@hugo:~$ lpstat
```

```
rsimms@hugo:~$ lpstat -p
```

```
printer HP_LaserJet_1320_series is idle.  enabled since Tue 08 May  
2012 08:46:45 PM PDT
```

```
rsimms@hugo:~$ lpstat -p -d
```

```
printer HP_LaserJet_1320_series is idle.  enabled since Tue 08 May  
2012 08:46:45 PM PDT
```

```
system default destination: HP_LaserJet_1320_series
```

*The -p option will show the available printers*

*The -d option will identify the default printer*

# CUPS

## lpstat command



### *On Opus*

```
/home/cis90/simben $ lpstat -p -d  
printer charlie disabled since Tue 26 Jan 2010 05:03:19 PM PST -  
    I don't really exist  
printer hplaser disabled since Tue 26 Jan 2010 04:58:14 PM PST -  
    Out of paper  
system default destination: charlie
```

*There are two "pretend" printers named charlie and hplaser on Opus*

# CUPS

## lp and lpr commands



*Use **lp** (or **lpr**) to print files*

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

```
/home/cis90/simben $ lp -d hplaser lab10  
request id is hplaser-6 (1 file(s))
```

*With **lp**, use the **-d** option to manually select a printer*

```
/home/cis90/simben $ lpr lab10
```

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

*With **lpr**, use the **-P** option to manually select a printer*

# CUPS

## lp and lpr commands



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

*Note that both lp and lpr will read from stdin.*

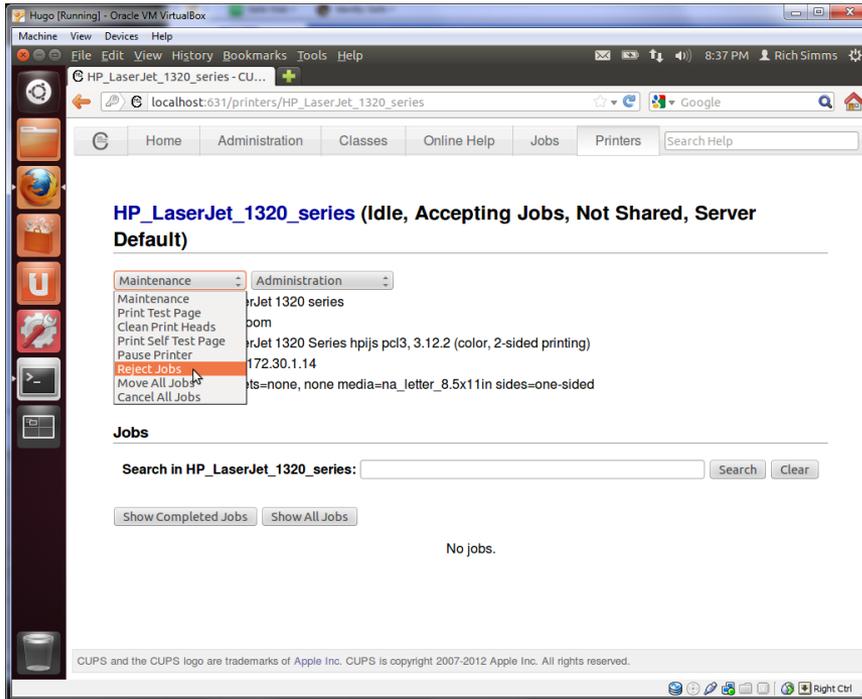
*This allows output from another command to be piped in*



# Managing Print Jobs

# CUPS

## Rejecting Jobs



*Clicking the **Reject Jobs** selection 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 ~]#
```

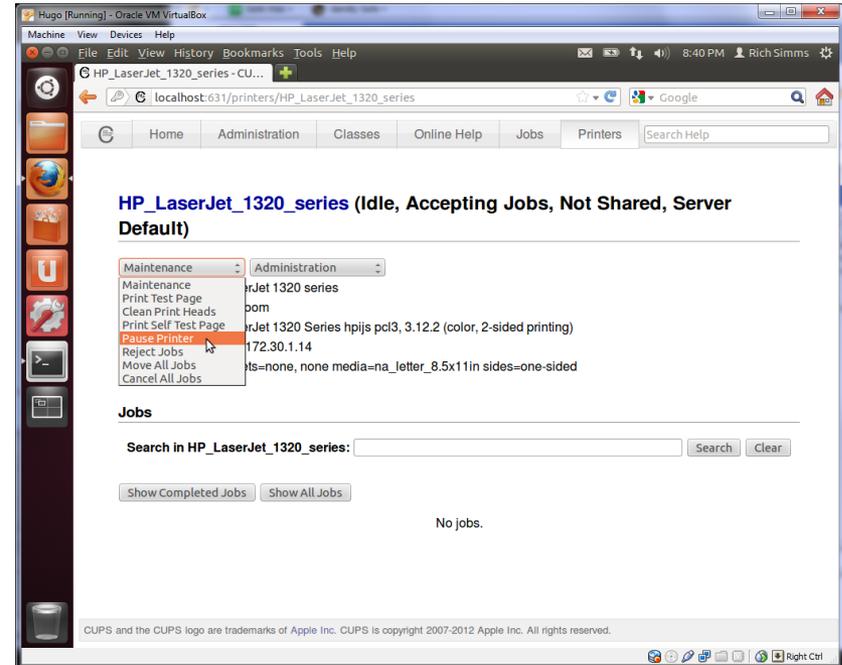
# CUPS

## Pausing the Printer

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

```
[root@benji ~]# lpq
hp7550 is not ready
Rank    Owner    Job      File(s)
Total Size
1st     root     22      myfile
1024 bytes
```

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



*Clicking the **Pause Printer** selection 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 **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:

|                                                         |                                     |
|---------------------------------------------------------|-------------------------------------|
| <a href="http://hostname:631">http://hostname:631</a>   | - CUPS web based management utility |
| <a href="http://hostname:9100">http://hostname:9100</a> | - 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



# Refresh



# Process Life Cycle

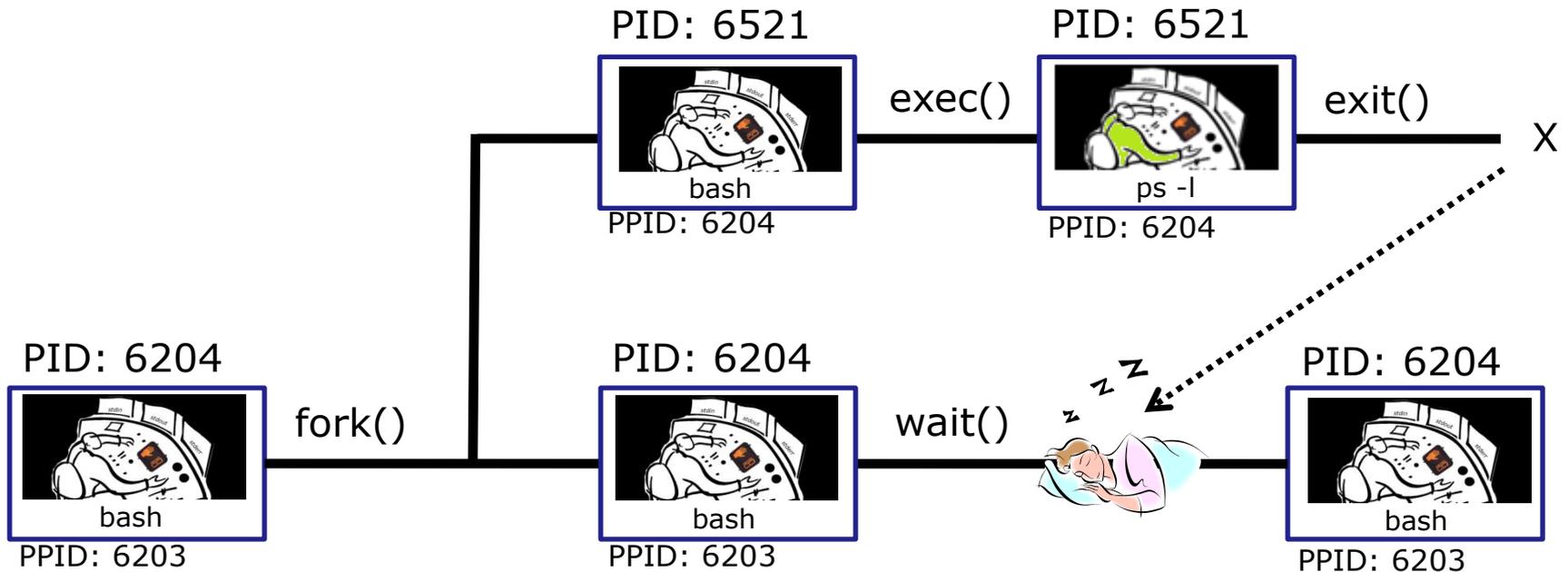
## An example process

*Running the **ps** command*

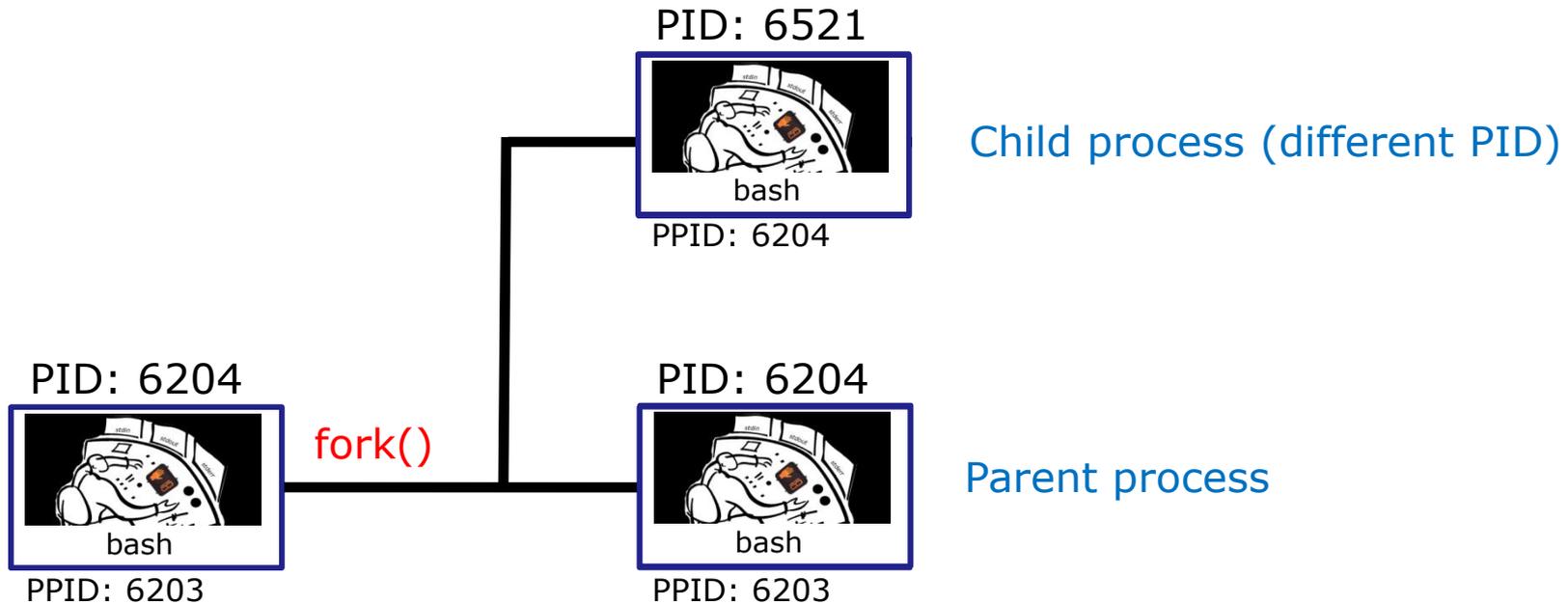
```
[rsimms@opus ~]$ ps -l
F S  UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  201  6204  6203  0  75   0  -   1165  wait  pts/6      00:00:00 bash
0 R  201  6521  6204  0  77   0  -   1050  -     pts/6      00:00:00 ps
```

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

# Example Process Lifecycle



# Example Process Lifecycle

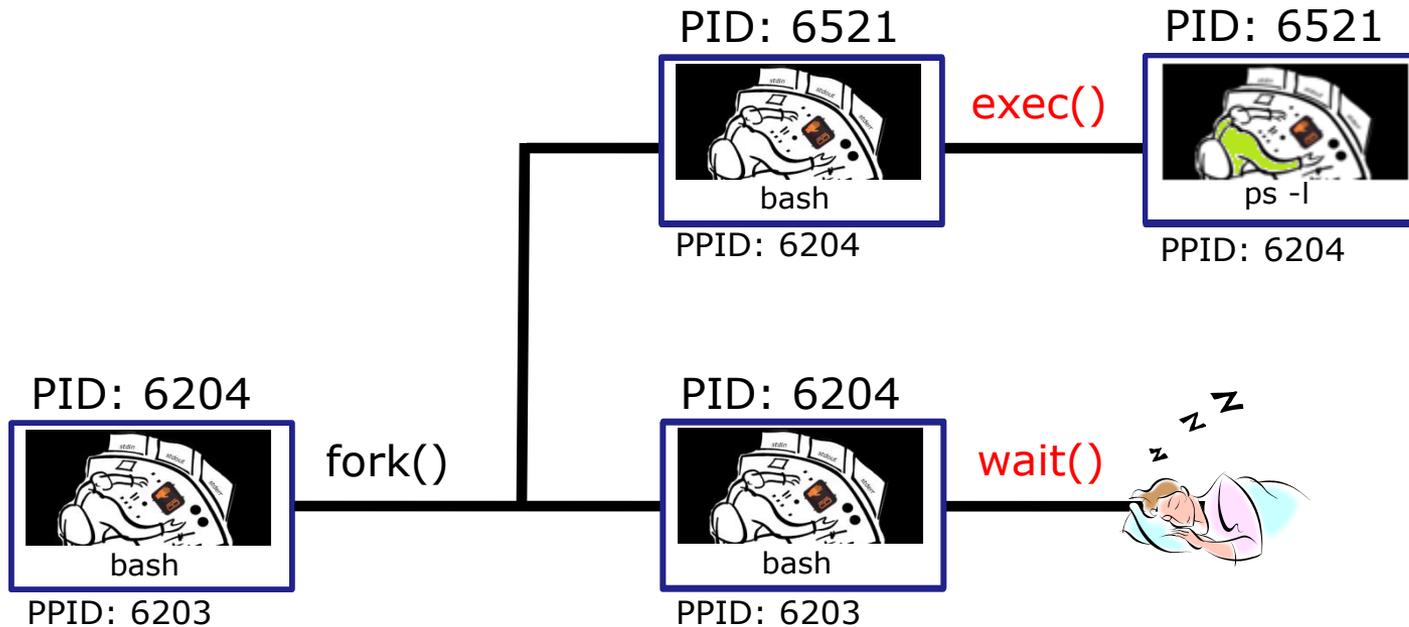


1) When a program is loaded into memory a new process must be created.

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

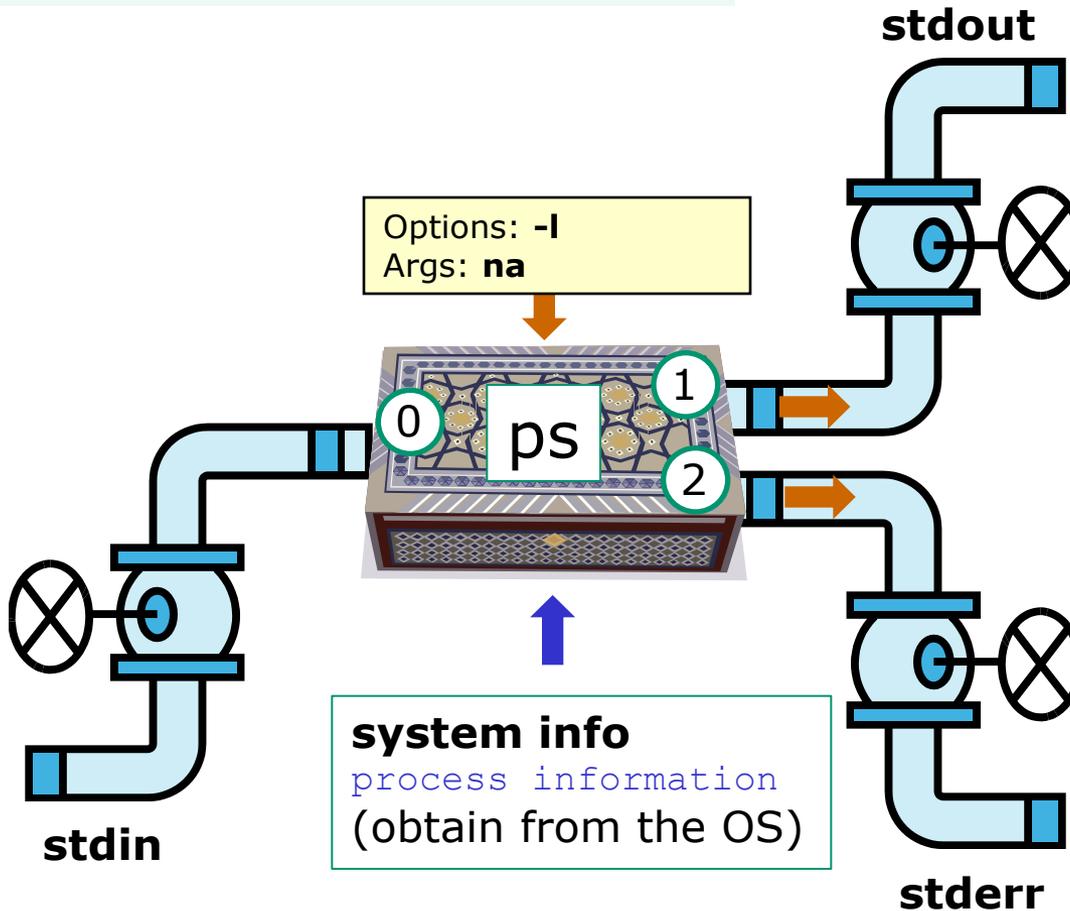


2) An `exec` system call is issued to overlay the **child** process with the instructions of the requested command (in this case the **ps** command). The new instructions then are executed.

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
```

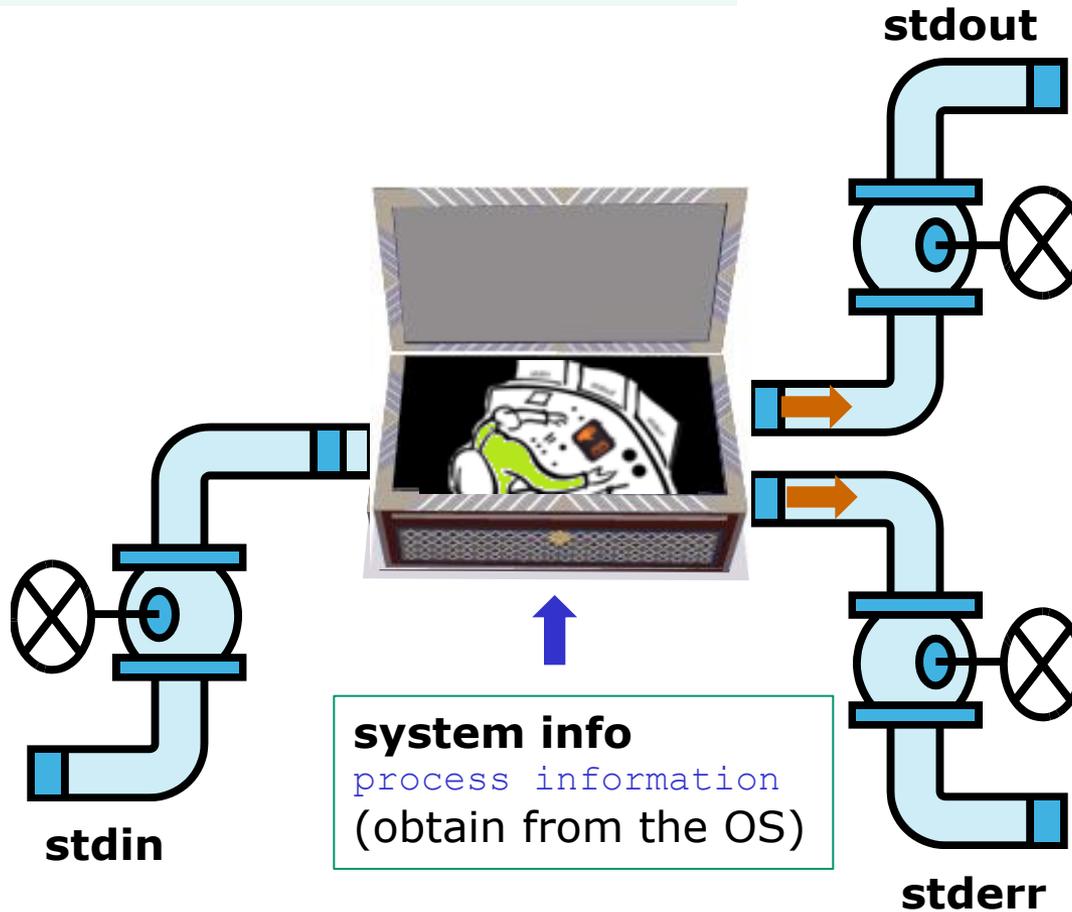


```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 201 6204 6203 0 75 0 - 1165 wait pts/6 00:00:00 bash
0 R 201 6521 6204 0 77 0 - 1050 - pts/6 00:00:00 ps
```

*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



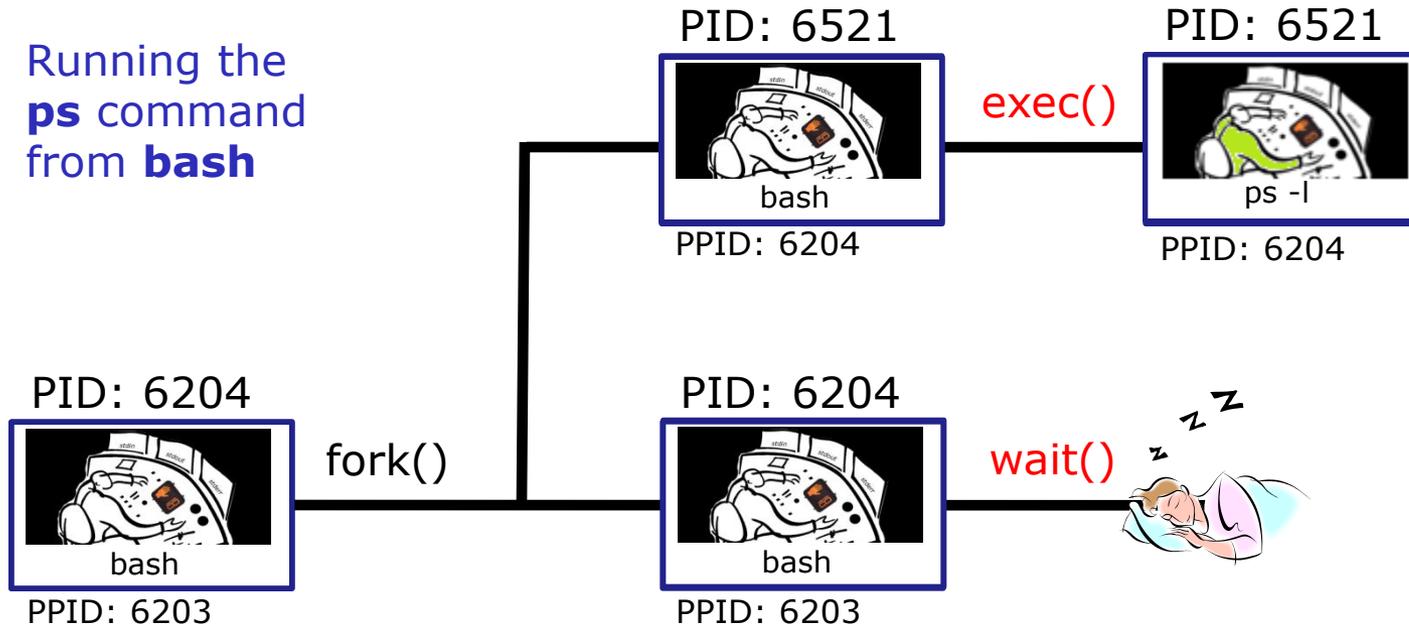
| F | S | UID | PID  | PPID | C | PRI | NI | ADDR | SZ   | WCHAN | TTY   | TIME     | CMD  |
|---|---|-----|------|------|---|-----|----|------|------|-------|-------|----------|------|
| 0 | S | 201 | 6204 | 6203 | 0 | 75  | 0  | -    | 1165 | wait  | pts/6 | 00:00:00 | bash |
| 0 | R | 201 | 6521 |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



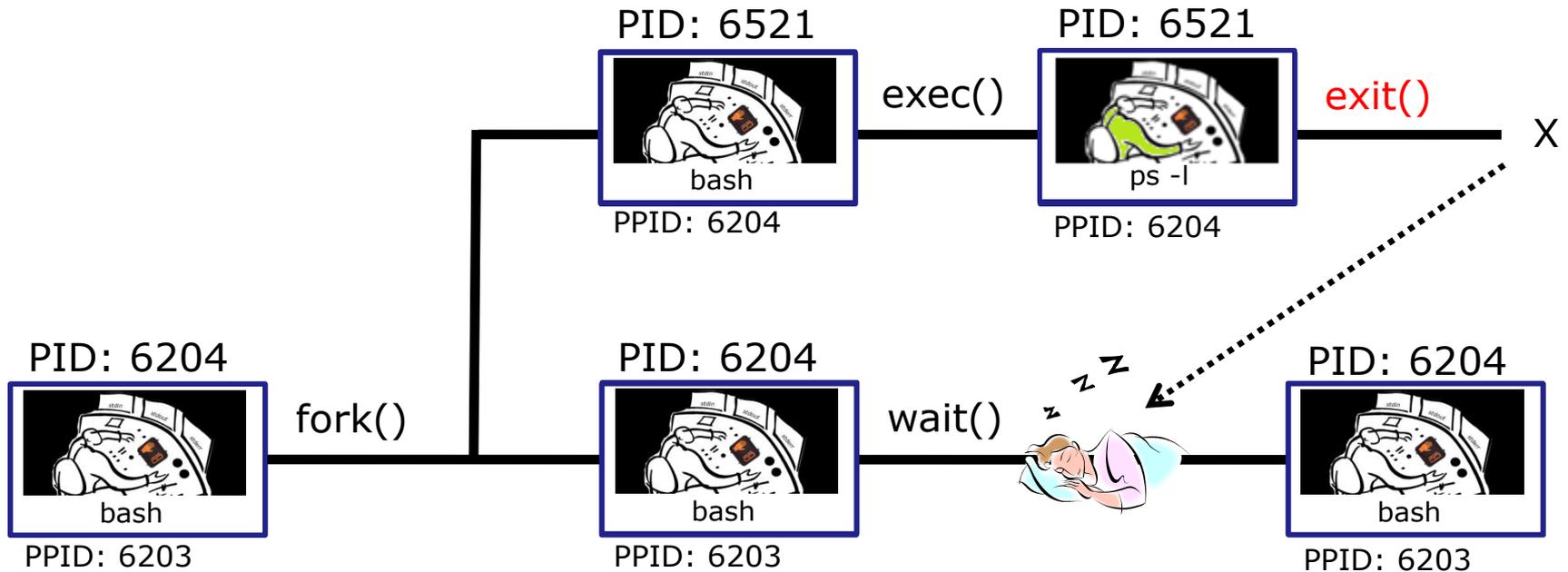
```
[rsimms@opus ~]$ ps -l
```

| 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.

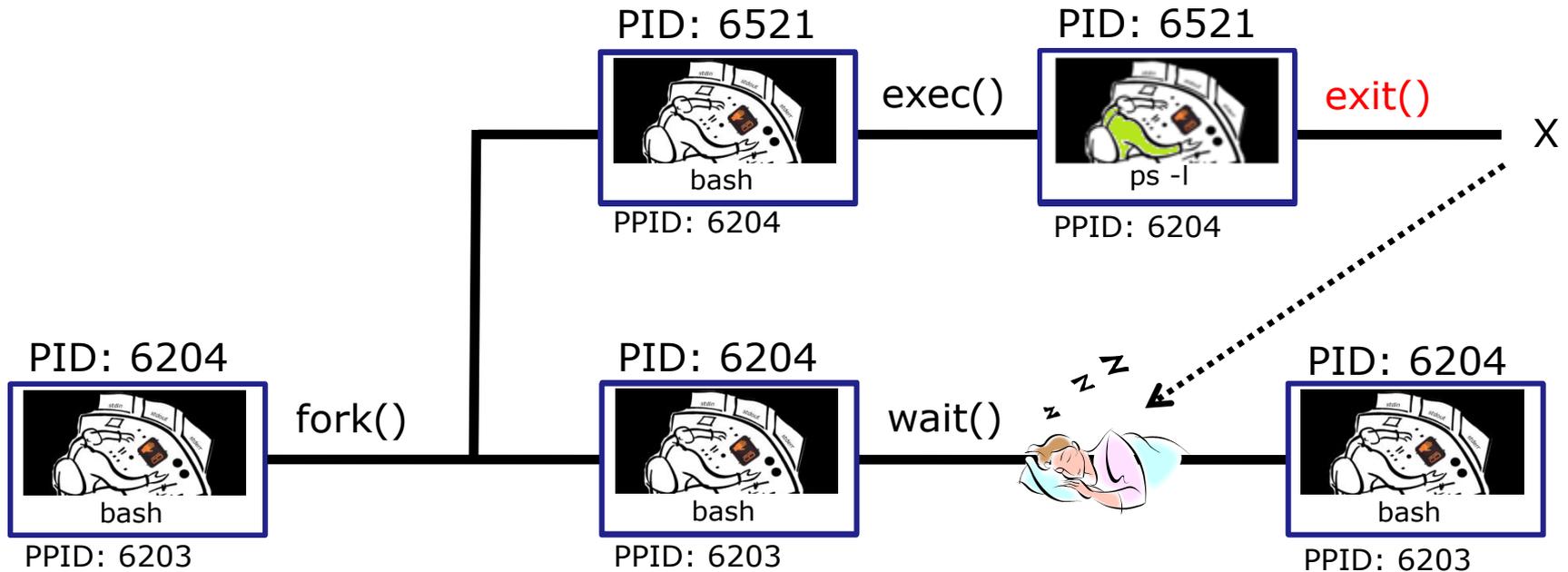
# Process Lifecycle



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



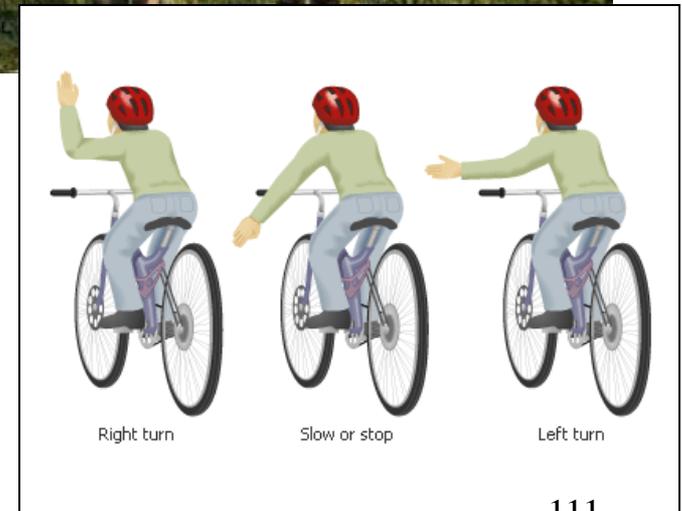
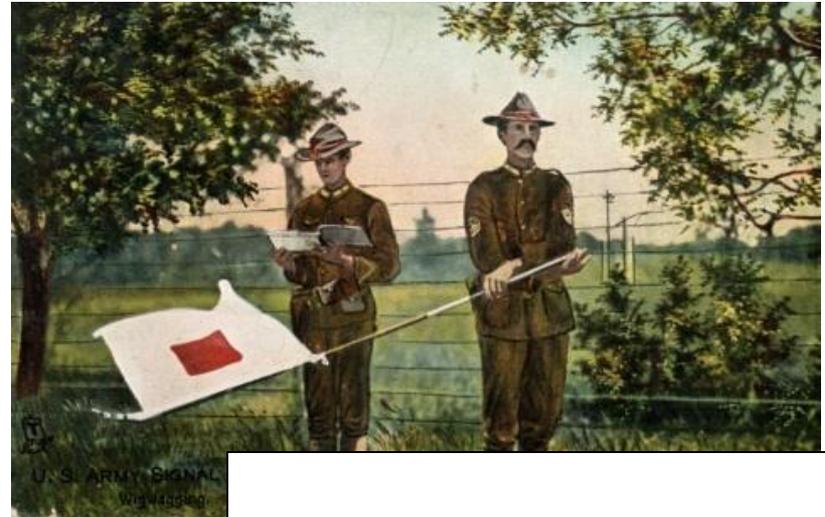
3) If the **parent** process were to die before the **child**, the zombie will become an **orphan**. Fortunately the init process will adopt any orphaned **zombies**.

# Signals

# Signals

PLATE 4

| COMMERCIAL CODE SIGNALS                                                                                                                                                                                                                                                                       |                                                    |                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <p>EXAMPLES OF THE SEVERAL HOISTS<br/>WHICH CAN BE MADE HAVING TWO, THREE, OR FOUR FLAGS.<br/>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 its 3rd occurrence, it must begin or be in the 3rd Hoist.</p> |                                                    |                                                                                                     |
| URGENT & IMPORTANT SIGNALS                                                                                                                                                                                                                                                                    |                                                    | COMPASS SIGNALS 3 FLAGS                                                                             |
| CODE FLAG OVER 1 FLAG OR 2 FLAG SIGNALS                                                                                                                                                                                                                                                       |                                                    |                                                                                                     |
| CODE FLAG<br>P<br>"I Am about to Sail"                                                                                                                                                                                                                                                        | A<br>"Do Not"                                      | A<br>Q<br>E<br>N 1/2 E<br>S 3/4 W                                                                   |
| LATITUDE & LONGITUDE SIGNALS                                                                                                                                                                                                                                                                  |                                                    | CODE FLAG OVER 2 FLAGS                                                                              |
| CODE FLAG<br>A<br>O<br>12° Latitude                                                                                                                                                                                                                                                           | GENERAL SIGNAL<br>Q<br>OR H<br>X<br>North Latitude | CODE FLAG<br>E<br>H<br>23° Longitude                                                                |
| NUMERAL TABLE                                                                                                                                                                                                                                                                                 |                                                    | GENERAL VOCABULARY                                                                                  |
| CODE FLAG UNDER 2 FLAGS<br>Y<br>S<br>CODE FLAG<br>10,000                                                                                                                                                                                                                                      | 3 FLAG SIGNAL<br>I<br>X<br>K<br>Tons of Coal       | GEOGRAPHICAL SIGNALS ALPHABETICAL ORDER.<br>4 FLAG SIGNAL<br>A<br>E<br>Y<br>Z<br>Glasgow, Scotland. |
| ALPHABETICAL SPELLING TABLE                                                                                                                                                                                                                                                                   |                                                    | NAMES OF VESSELS FROM CODE LIST.                                                                    |
| 1, 2, 3 OR 4 FLAG SIGNALS<br>J<br>O<br>H<br>N<br>John                                                                                                                                                                                                                                         | 4 FLAG SIGNALS<br>C<br>B<br>D<br>N<br>Abb<br>at    | 4 FLAG SIGNAL<br>H<br>C<br>L<br>B<br>Grays of Glasgow<br>1058 Tons No 52636                         |



# 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

# 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) <b>Ctrl-C</b>                          |
| SIGQUIT | 3  | Terminal quit (POSIX) <b>Ctrl-\</b>                              |
| 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) <b>can't be caught or ignored</b>                   |
| 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) <b>default kill signal when not specified</b> |

*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) <b>Ctrl-Z or Ctrl-F</b> |
| 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'
```

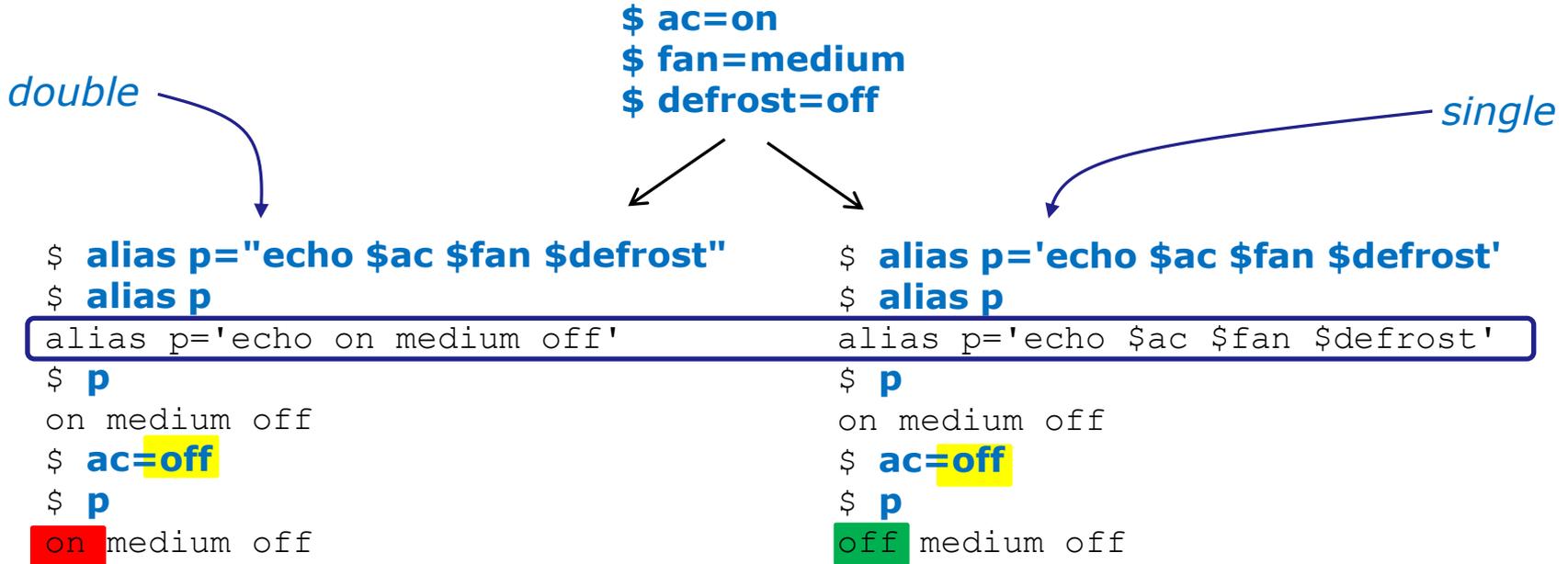
*Using the type or alias command to show an existing alias*

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

*Using unalias command to remove an alias*

# alias command

strong (') or weak (") quote marks



*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:* .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

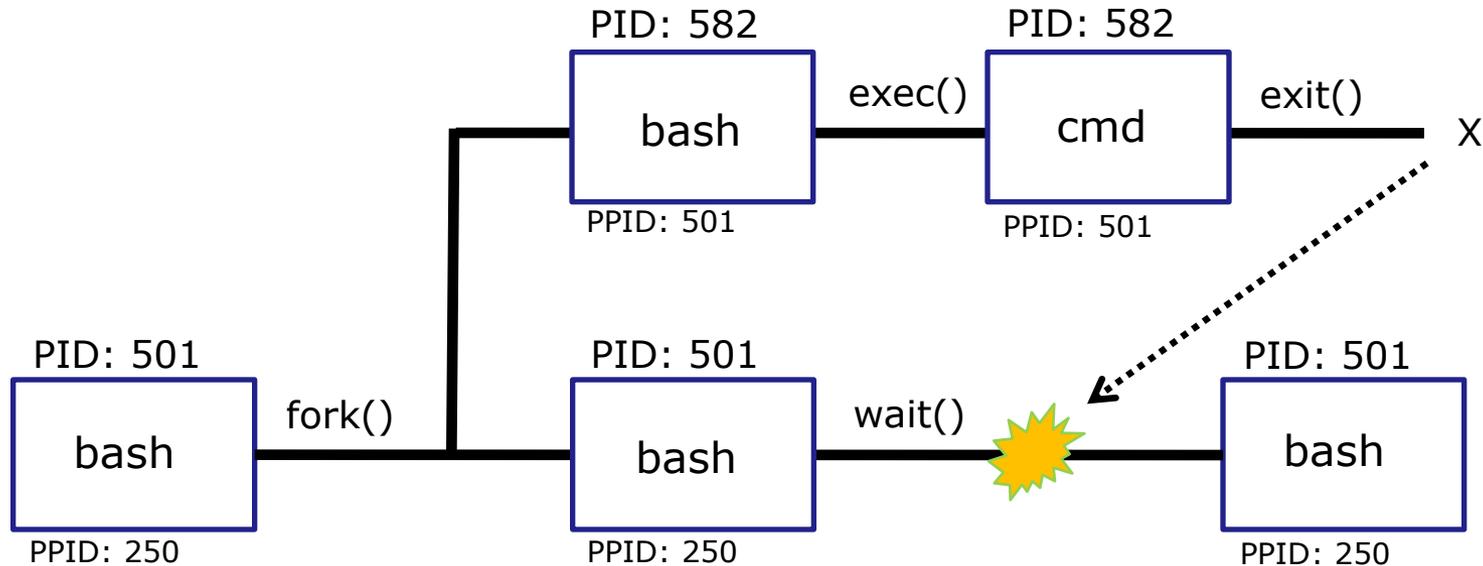
## export command – show all exported variables

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

```
declare -x BASH_ENV="/home/cis90/roddyduk/.bashrc"
declare -x CVS_RSH="ssh"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/cis90/roddyduk"
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="roddyduk"
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/roddyduk"
declare -x OLDPWD
declare -x
PATH="/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/roddyduk/../../bin:/home/cis90/roddyduk/bin:."
declare -x PWD="/home/cis90/roddyduk"
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="roddyduk"
declare -x USERNAME=""
[roddyduk@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

## **/etc/profile** (all)

- adds root's special path

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

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

## **.bash\_profile** (user specific)

- adds user's bin to path

## **.bashrc** (user specific)

- add aliases here

## **/etc/bashrc** (all)

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

*only  
executed  
when  
logging in*

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

# .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

```

roddyduk@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***

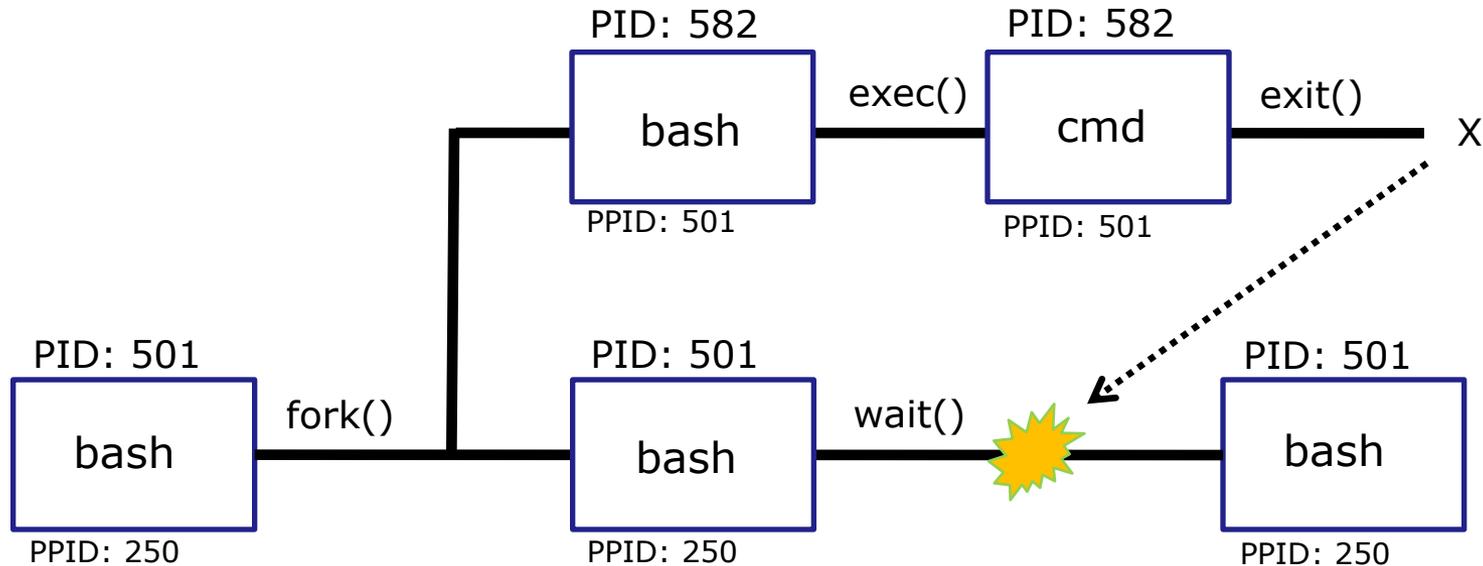
In this example, the commands in the file `shscript` are run by the parent shell, and therefore, any changes made to the environment will last for the duration of the login session.

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*

[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*

[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*

# CUPS

Administration - CUPS 1.2.4 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.0.25:631/admin/

Administration - CUPS 1.2.4 Using Network Printers - CUPS 1.2.4

## Administration

Home Administration Classes Documentation/Help Jobs Printers

### Printers

Add Printer Manage Printers

New Printers Found:

- Add This Printer HP Fax (no\_device\_found)

### Classes

Add Class Manage Classes

### Jobs

Manage Jobs

### Server

Edit Configuration File View Access Log View Error Log View Page Log

#### Basic Server Settings:

- Show printers shared by other systems
- Share published printers connected to this system
- Allow remote administration
- Allow users to cancel any job (not just their own)
- Save debugging information for troubleshooting

Change Settings

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

*To add in  
HP 1320N  
printer ...*

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

# CUPS

, "#", and space)'), 'Location:' (with a note: '(Human-readable location such as \"Lab 1\")'), and 'Description:' (with a note: '(Human-readable description such as \"HP LaserJet with Duplexer\")'). A 'Continue' button is located below the description field. The footer contains copyright information: '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.' The status bar at the bottom shows 'Done' and the IP address '192.168.0.25:631'."/>

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

**UNIX PRINTING SYSTEM** Add Printer

Home Administration Classes Documentation/Help Jobs Printers

### Add New Printer

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

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

**Description:**   
(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

*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")

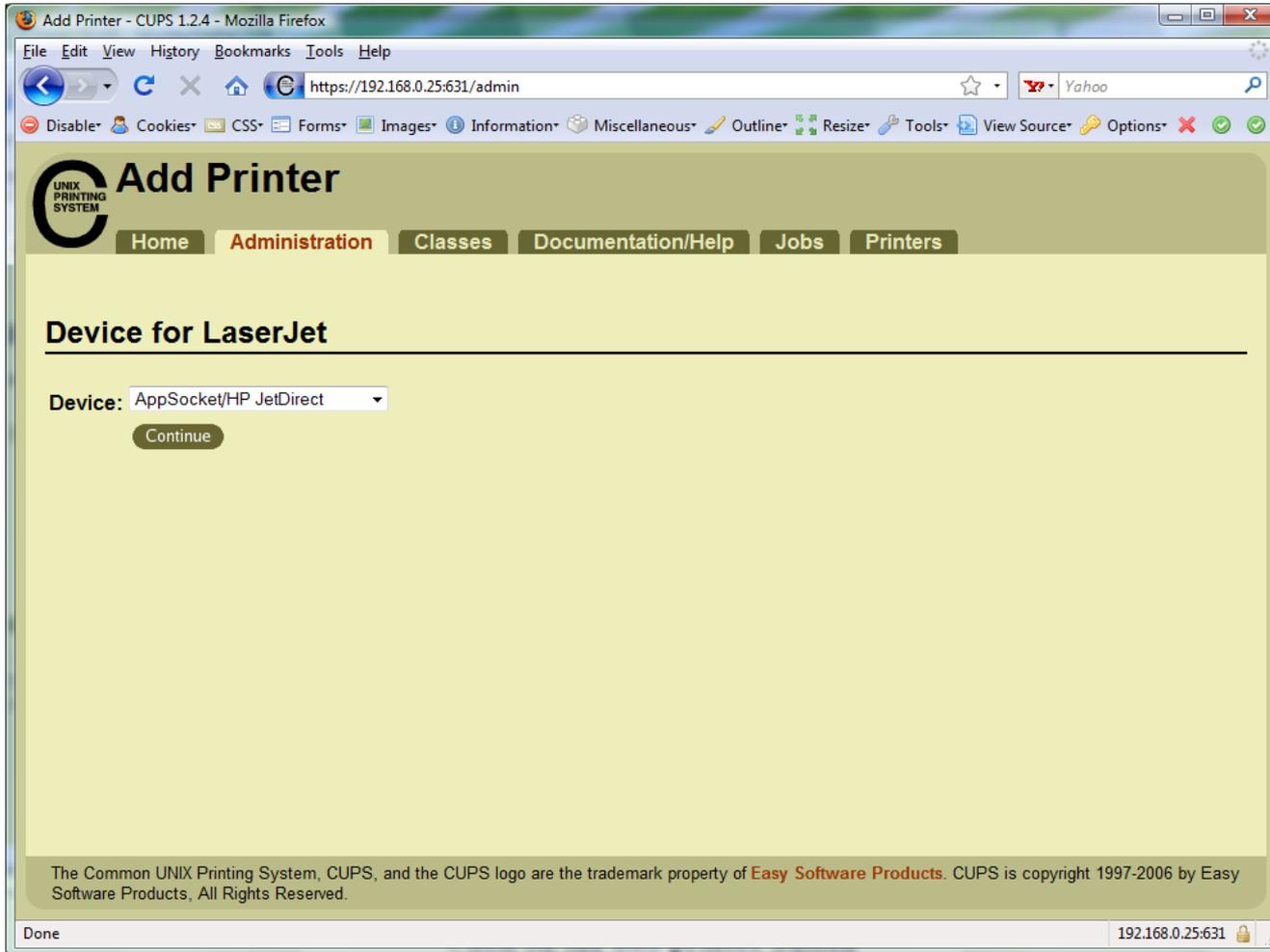
Continue

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*

Add Printer - CUPS 1.2.4 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.0.25:631/admin

Add Printer - CUPS 1.2.4 Using Network Printers - CUPS 1.2.4

## Add Printer

Home Administration Classes Documentation/Help Jobs Printers

### Device URI for LaserJet

Device URI:

Examples:

- http://hostname:631/ipp/
- http://hostname:631/ipp/port1
- ipp://hostname/ipp/
- ipp://hostname/ipp/port1
- lpd://hostname/queue
- socket://hostname
- socket://hostname:9100

See "Network Printers" for the correct URI to use with your printer.

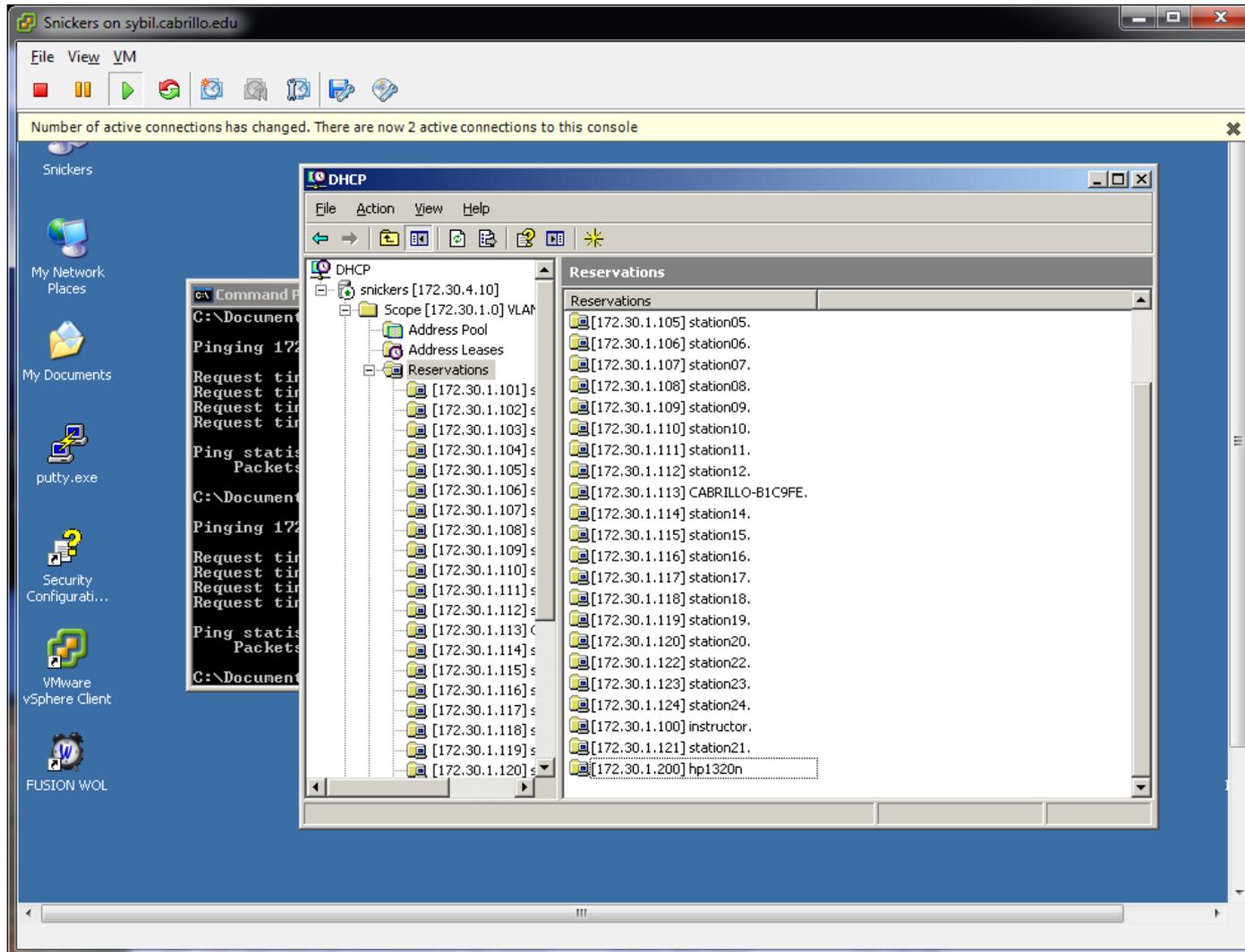
Continue

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

*This defines how to communicate with the printer*

# CUPS



*Room 2501: 172.30.1.200 for 0014389C595F*

# CUPS

The screenshot shows a web browser window displaying the HP LaserJet 1320 series configuration page. The address bar shows the URL `192.168.0.12/hp/jetdirect/`. The page title is `hp-laser / 192.168.0.12 hp LaserJet 1320 series`. The navigation tabs include **Information**, **Settings**, and **Networking**. The **Networking** tab is active, showing sub-tabs for **Misc. Settings**, **LPD Queues**, **Support Info**, and **Refresh Rate**. The **Misc. Settings** sub-tab is selected, displaying the following configuration options:

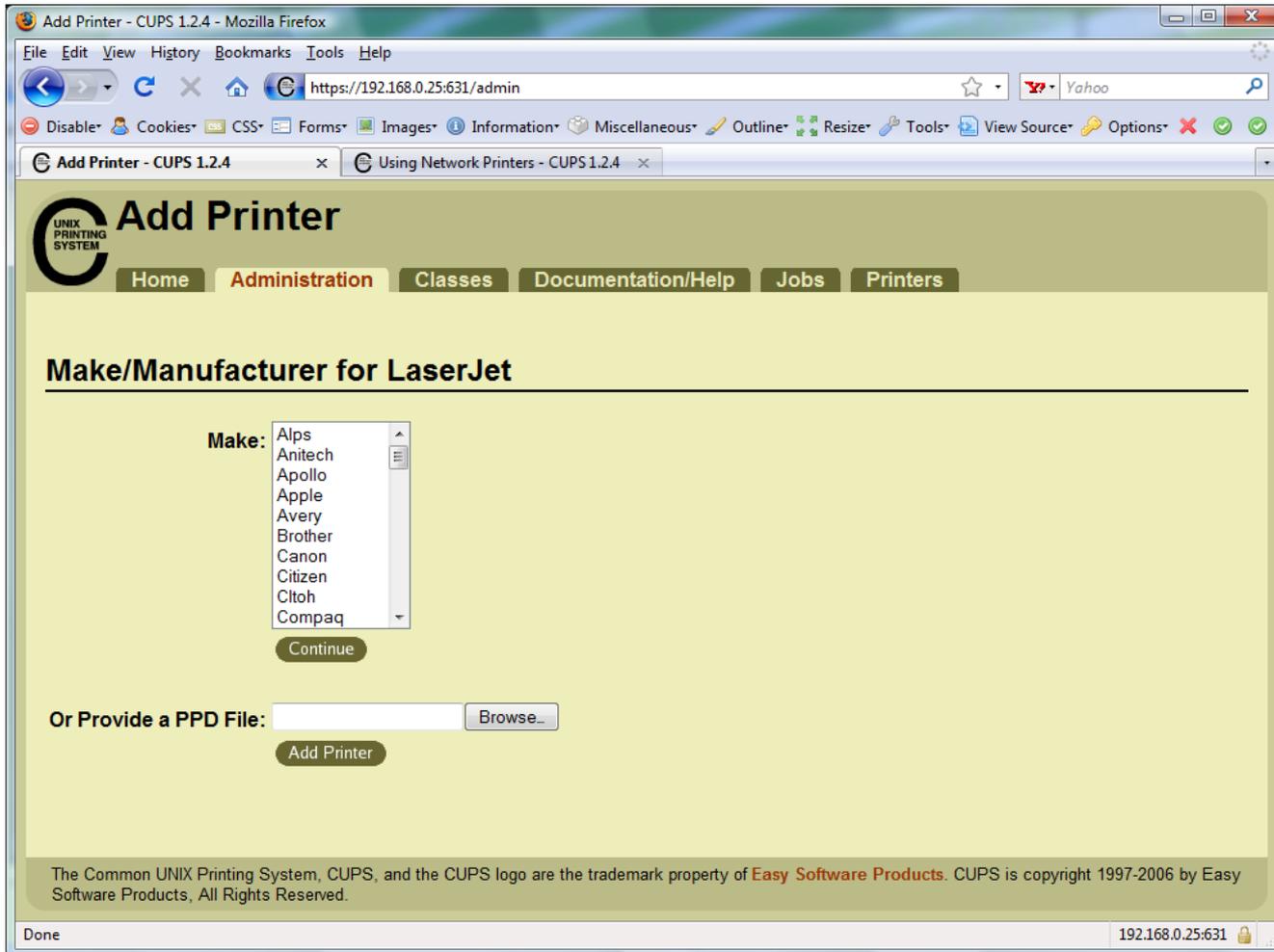
- Enabled Features:**
  - SLP Config
  - Telnet Config
  - mDNS
  - Multicast IPv4
  - 9100 Printing
  - LPD Printing
- Link settings:**
- Primary DNS Server:**
- Secondary DNS Server:**
- Locally Administered Address:**
- Syslog Facility:**
- Dynamic Raw Port Setting:**
  - Dynamic Raw Port 1:**
  - Dynamic Raw Port 2:**
- Printer Services:**
  - Disable listening on these ports:**
  - Streams:**
  - Datagrams:**

The left sidebar contains the following navigation links:

- CONFIGURATION**
  - Network Settings
  - Other Settings
  - Privacy Settings
- SECURITY**
  - Settings
  - Authorization
  - Mgmt. Protocols
- DIAGNOSTICS**
  - Network Statistics
  - Protocol Info
  - Configuration Page
- Other Links**
  - Help
  - Support
  - HP Home

*Room 2501: 172.30.1.14 for 0014389C595F*

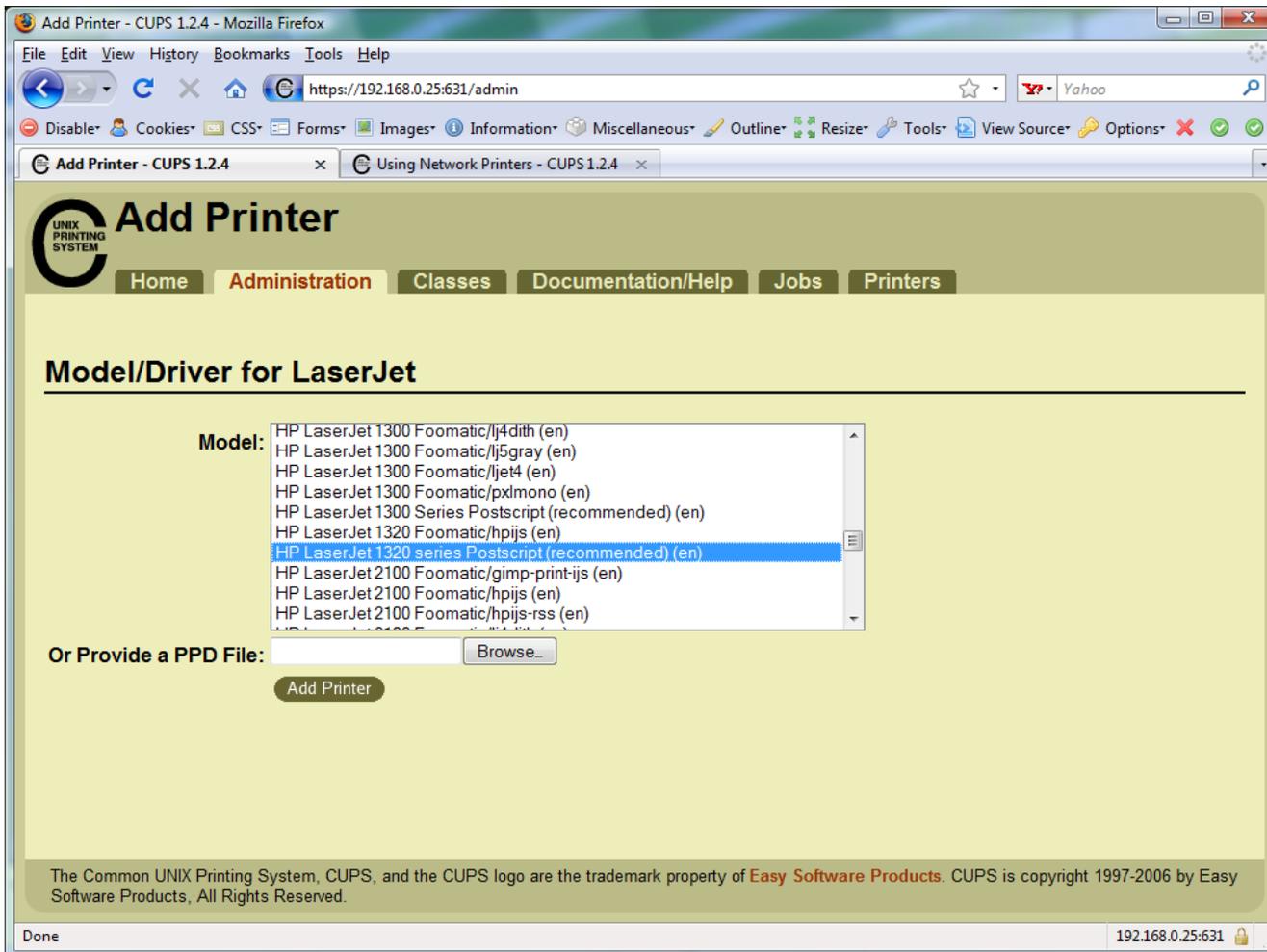
# CUPS



*(you will need to enter root's password)*

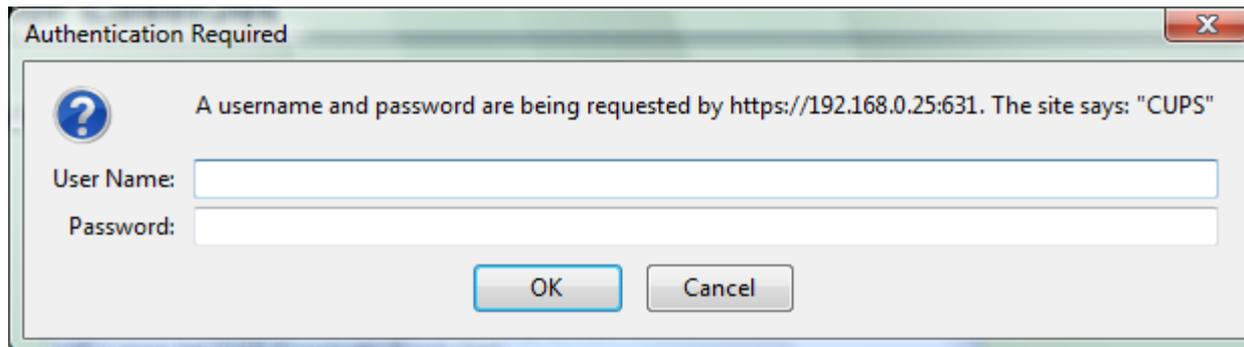
*Service will restart*

# CUPS



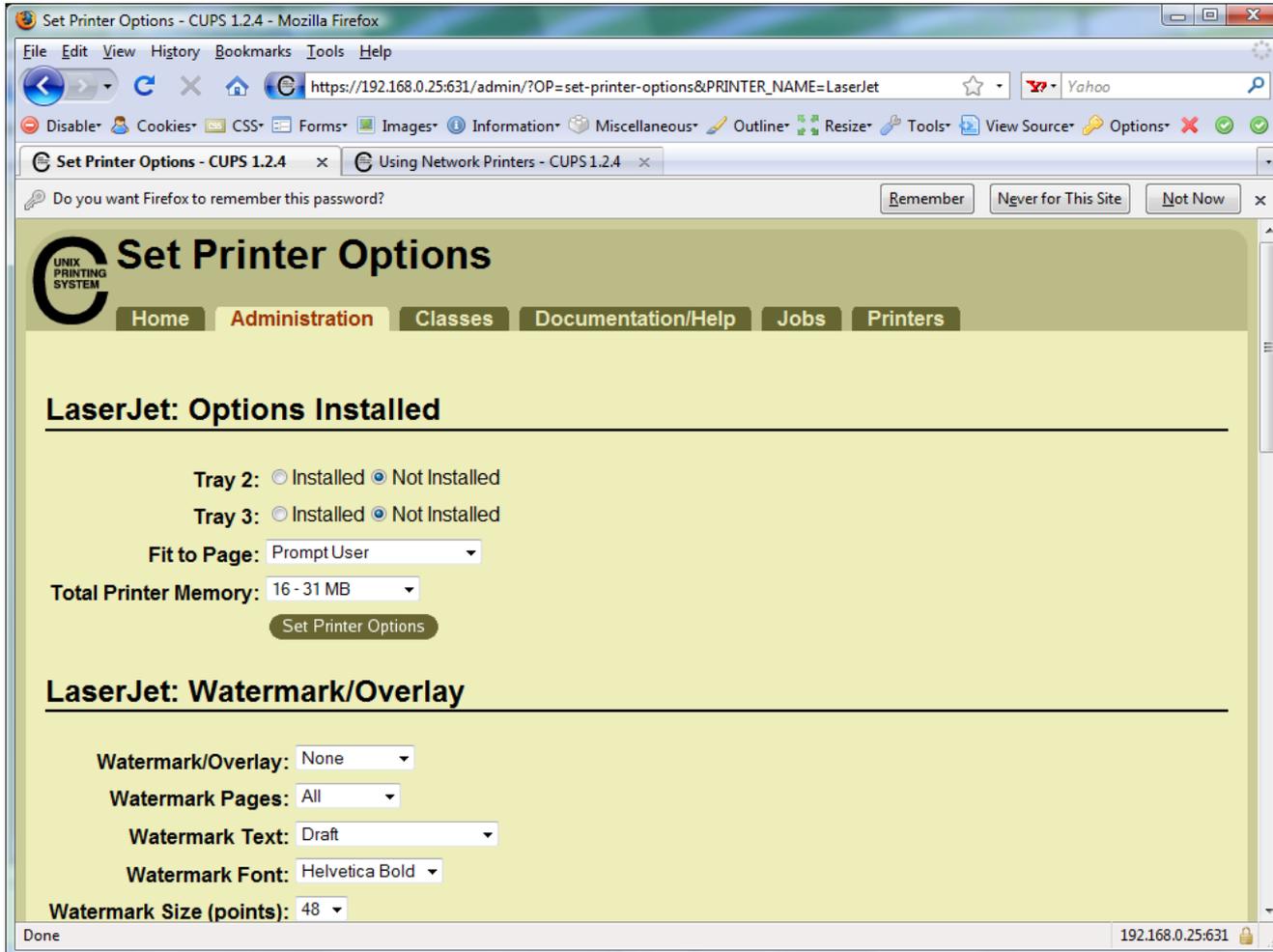
*We will choose hp  
LaserJet 1320  
series Postscript  
(recommended)  
(en)*

# CUPS



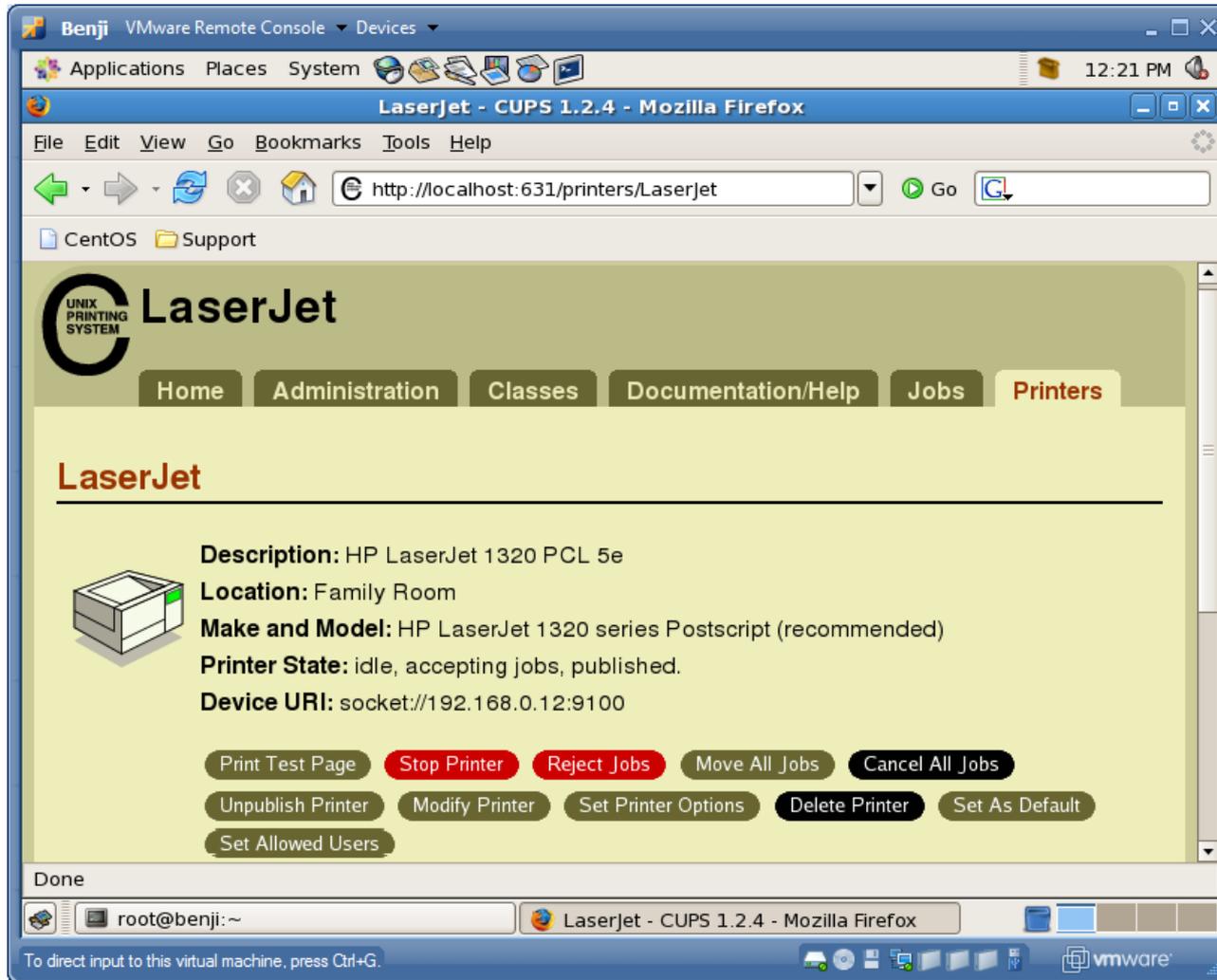
*To finally add the printer it will be necessary authenticate as root*

# CUPS



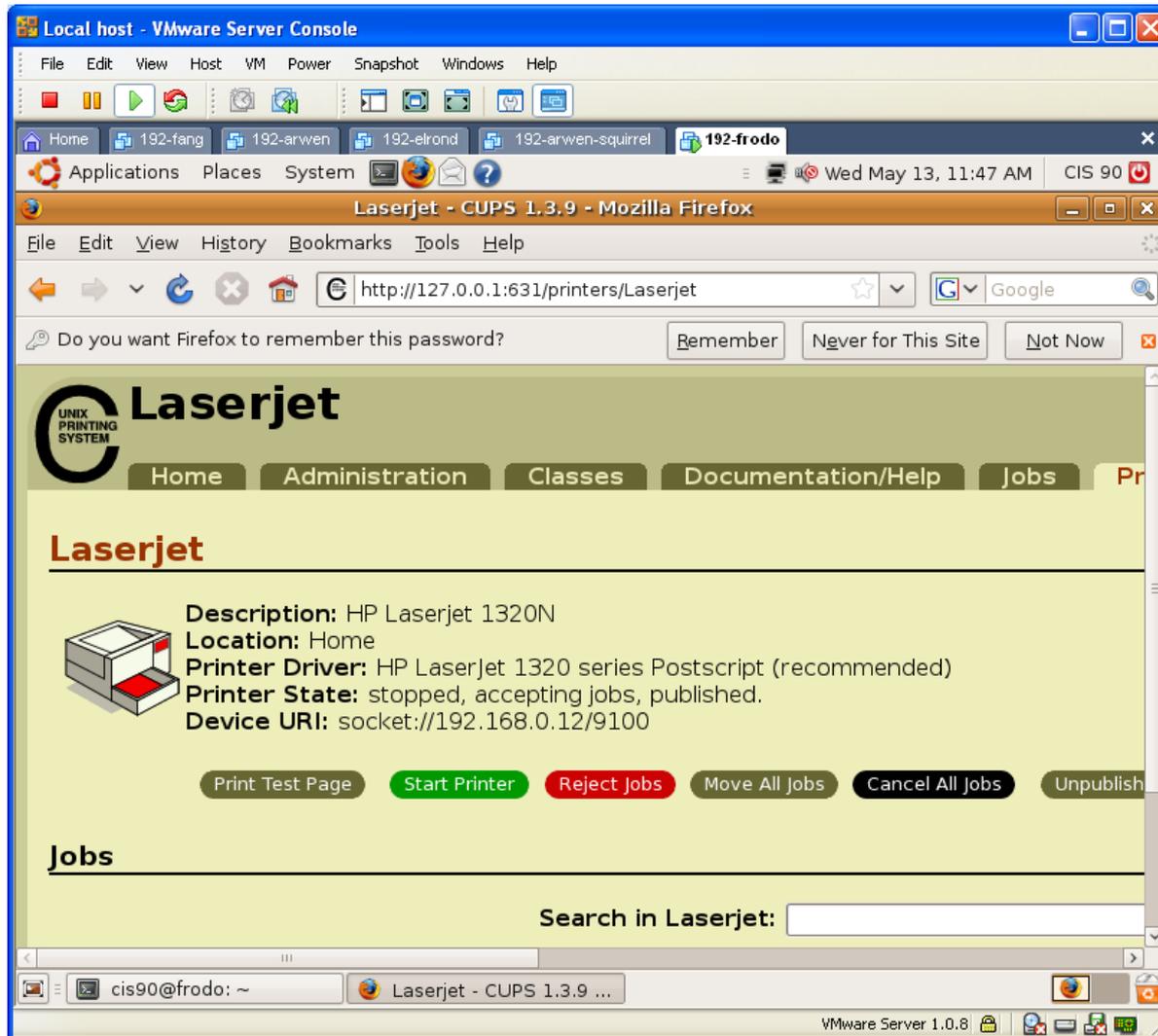
*Printer has  
been added*

# 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*

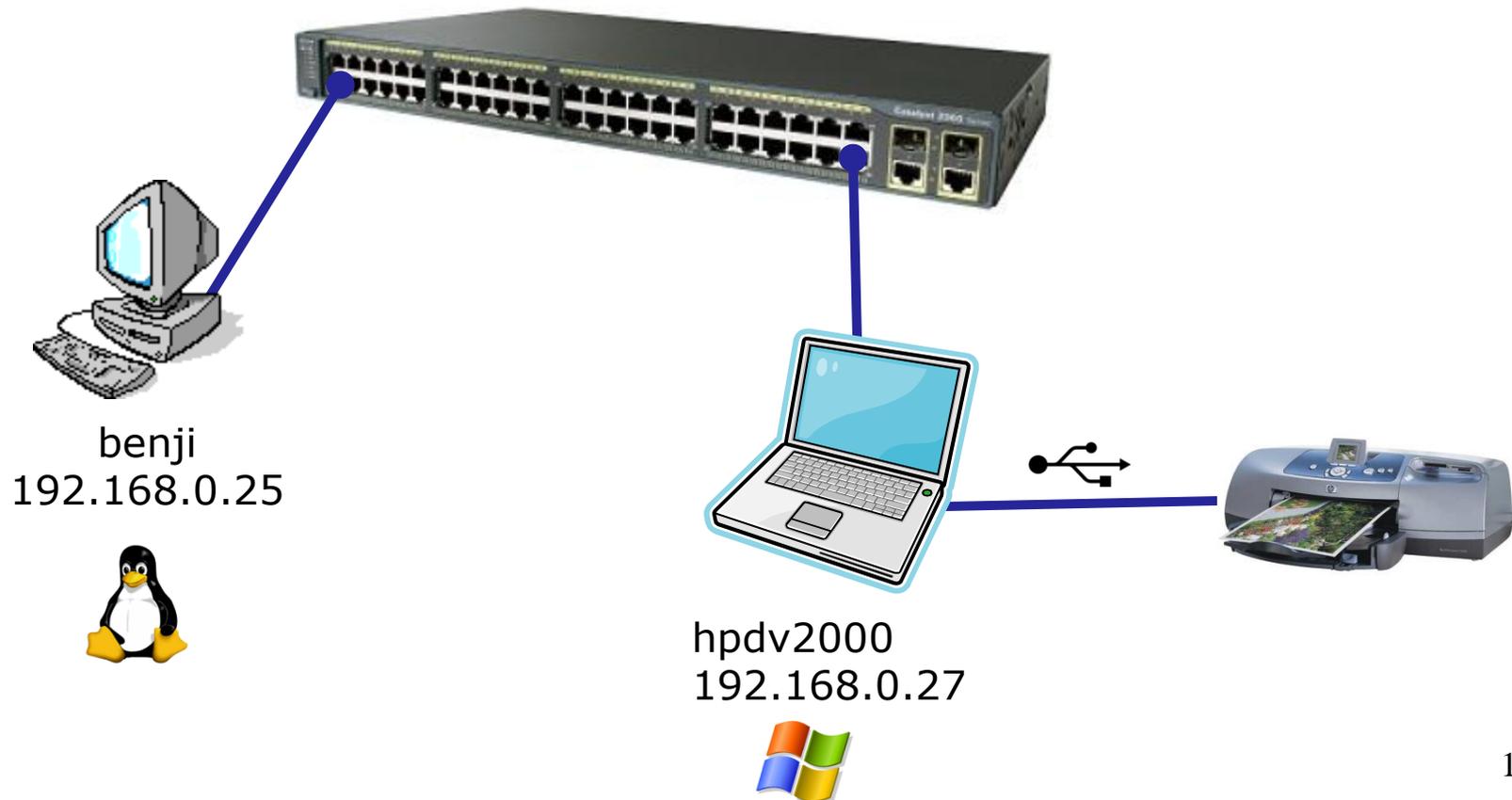
## Sidetrack – The previous 7550 "Hot Lips"



*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

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

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

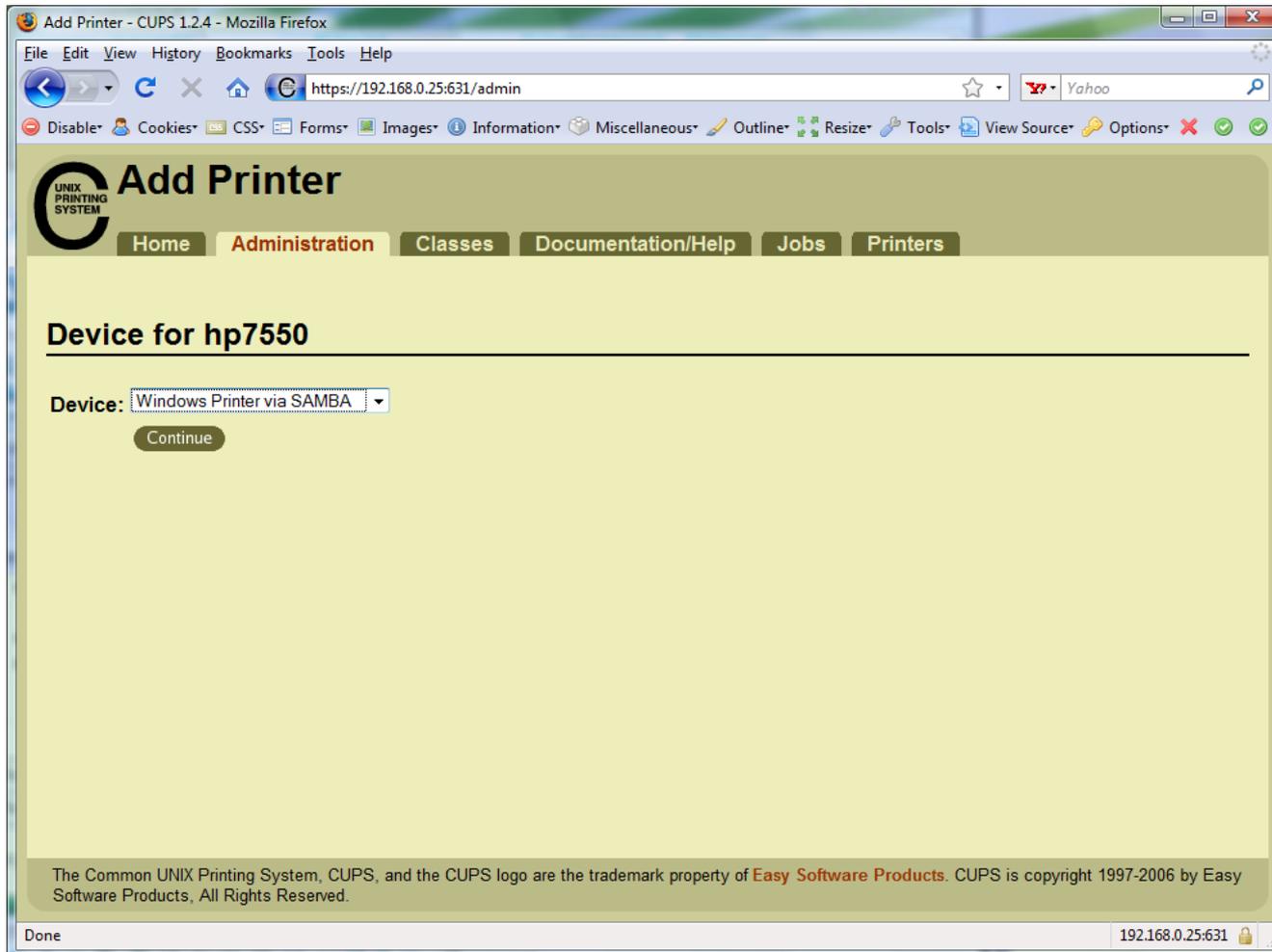
**Description:**   
(Human-readable description such as "HP LaserJet with Duplexer")

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

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

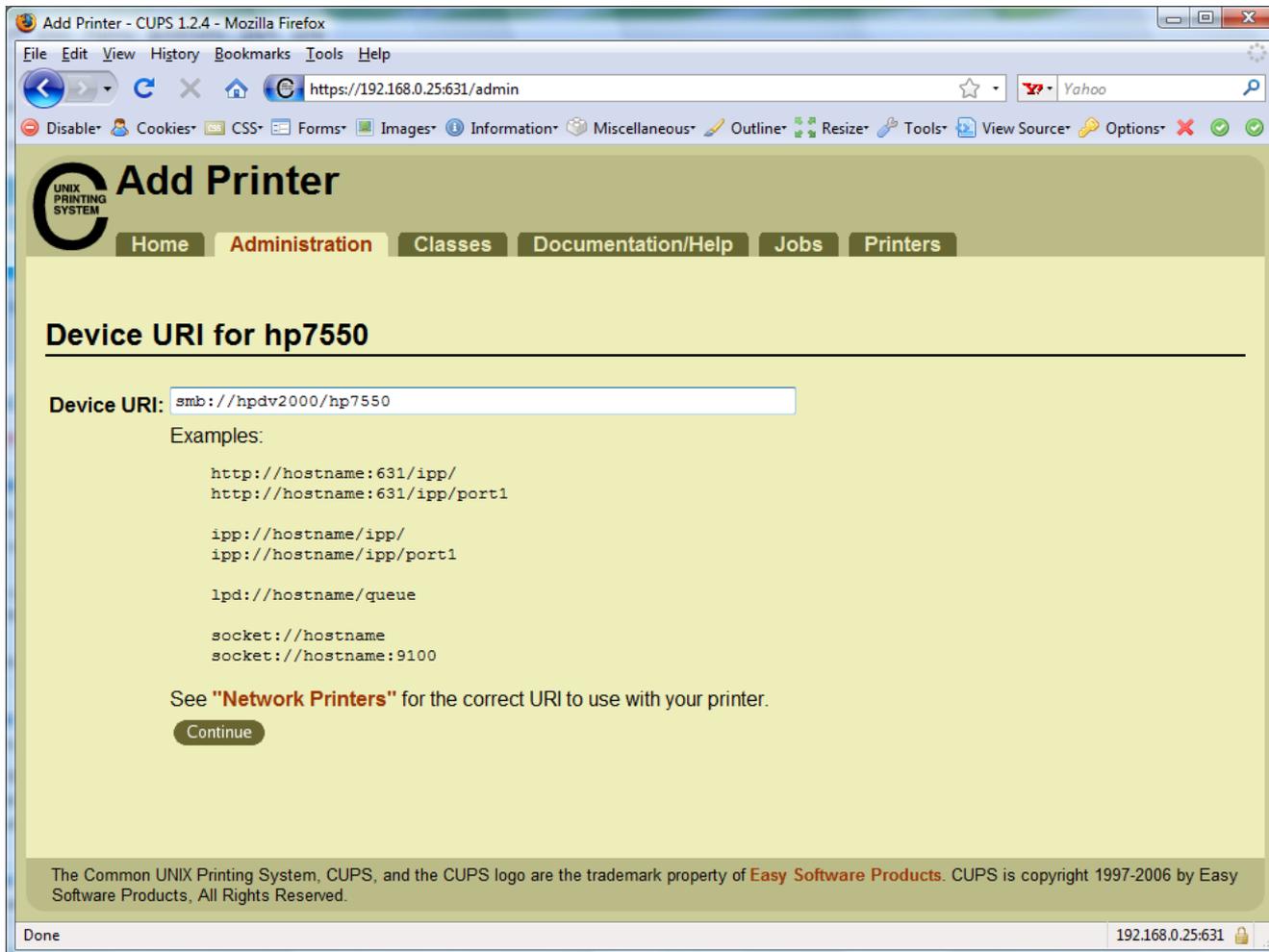
# CUPS



*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 administration interface in Mozilla Firefox, displaying the 'Add Printer' page. The 'Device URI' field is set to 'smb://hpdv2000/hp7550'. Below the field, there are examples of various URI schemes like http, ipp, lpd, and socket. A 'Continue' button is visible at the bottom of the page.

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 message: 'The command completed successfully.' and shows the prompt 'C:\Users\Administrator>'.

# CUPS

## *Ways to specify a Windows share*

### Username and password Not required

This machine is in the same workgroup

`smb://server/sharename`

This machine is in a different workgroup

`smb://workgroup/server/sharename`

### Username and password required

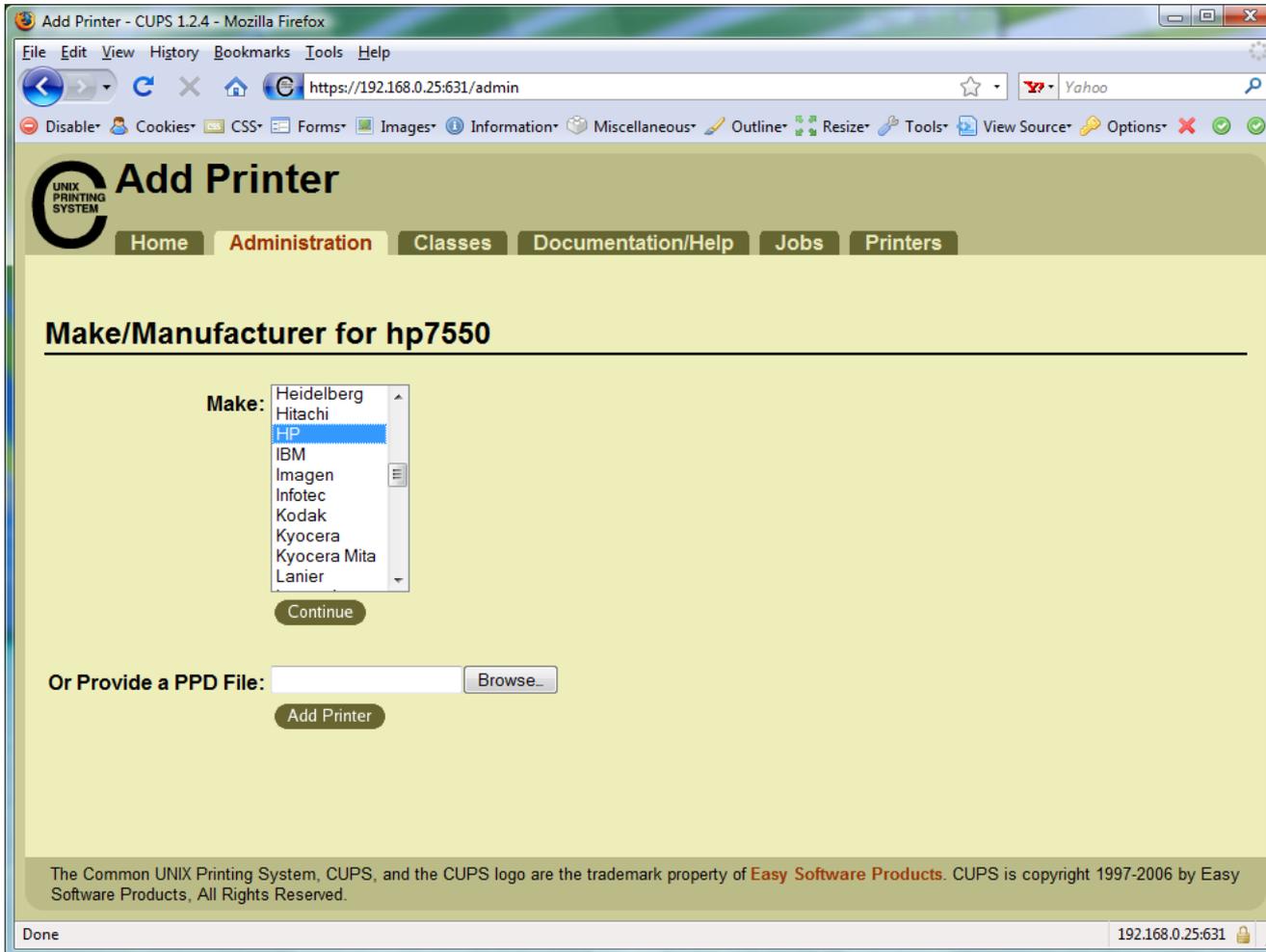
This machine is in the same workgroup

`smb://username:password@server/sharename`

This machine is in a different workgroup

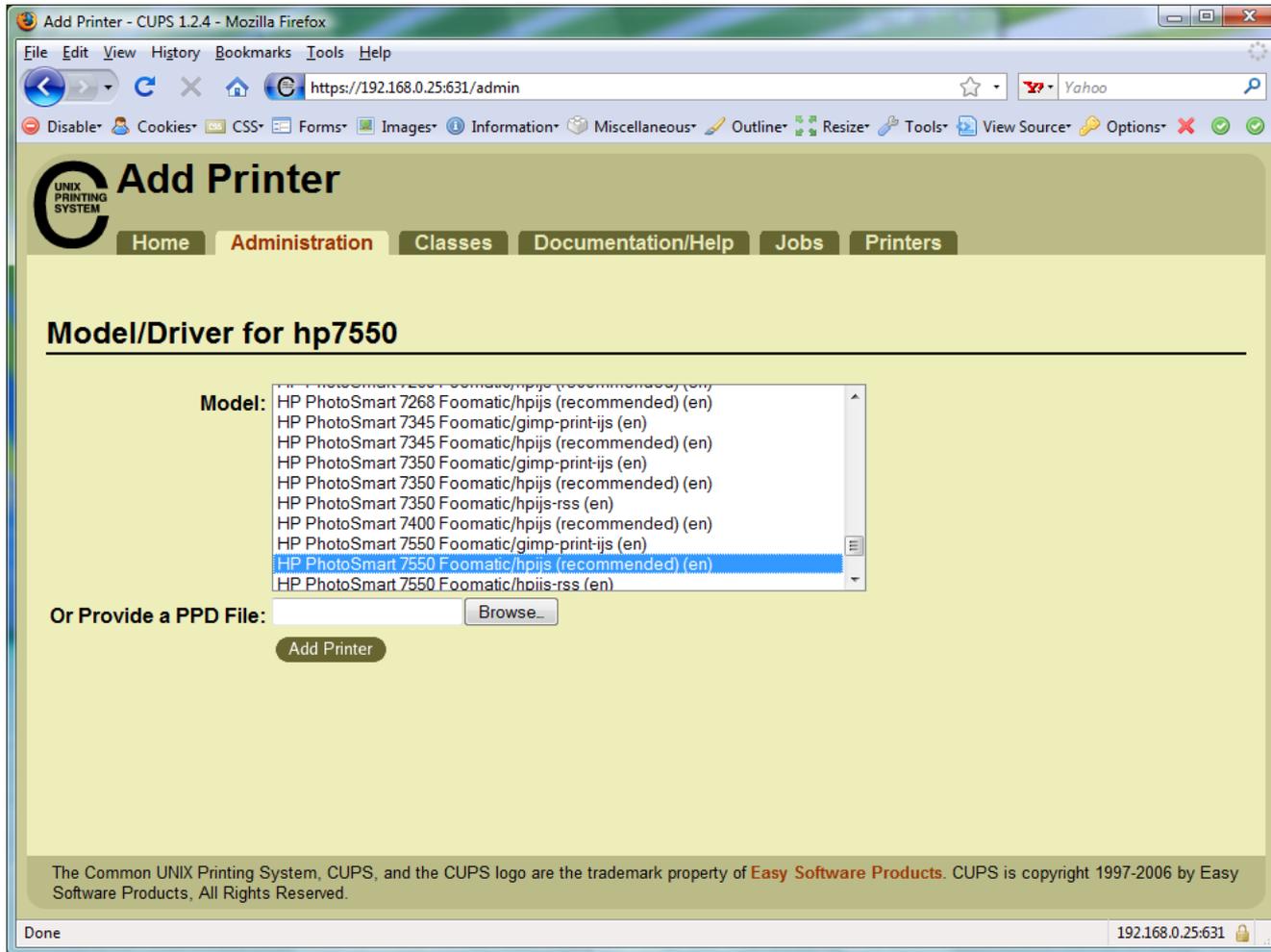
`smb://username:password@workgroup/server/sharename`

# CUPS



*Select make  
of printer*

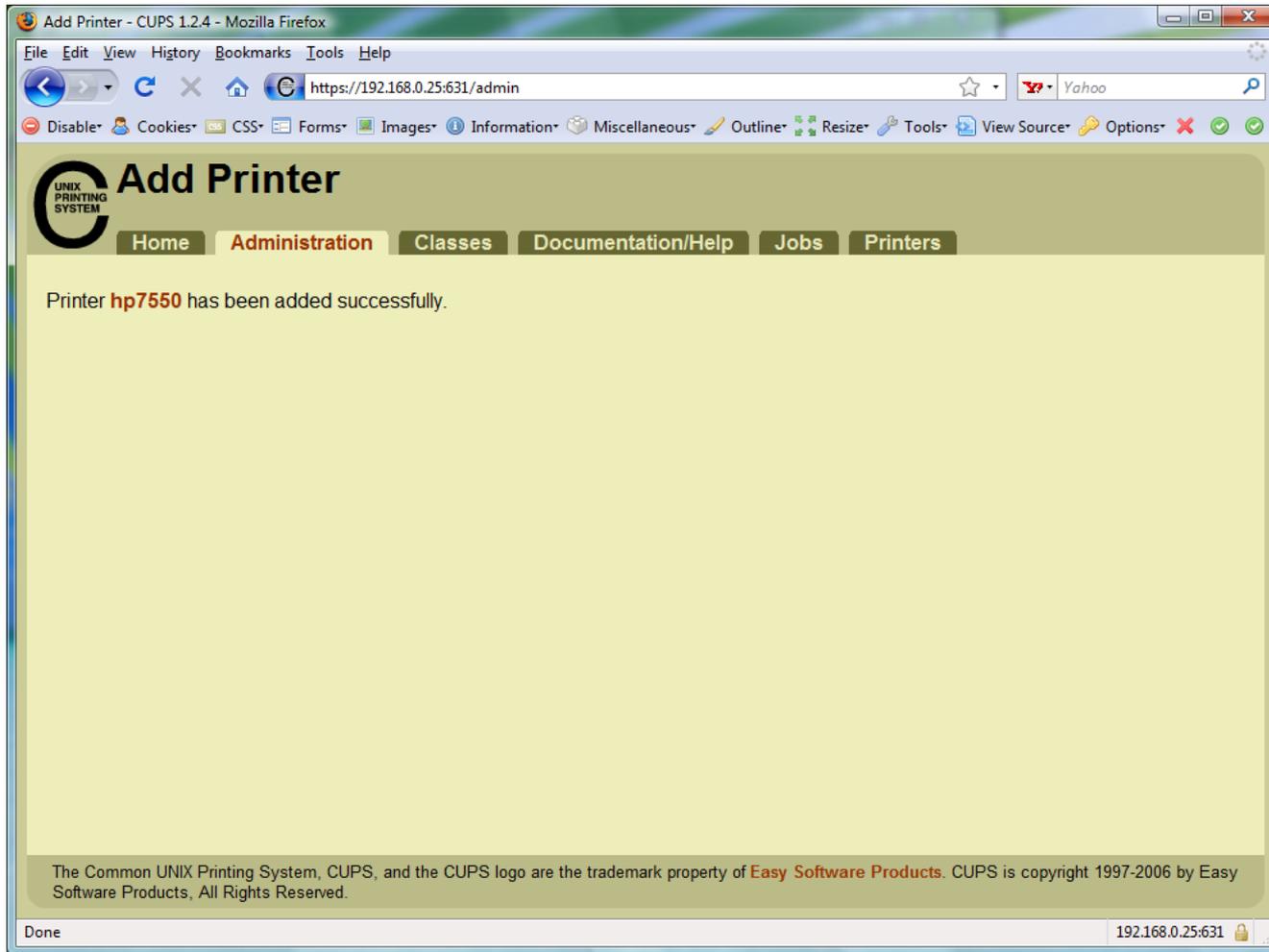
# CUPS



*Select model of  
printer*

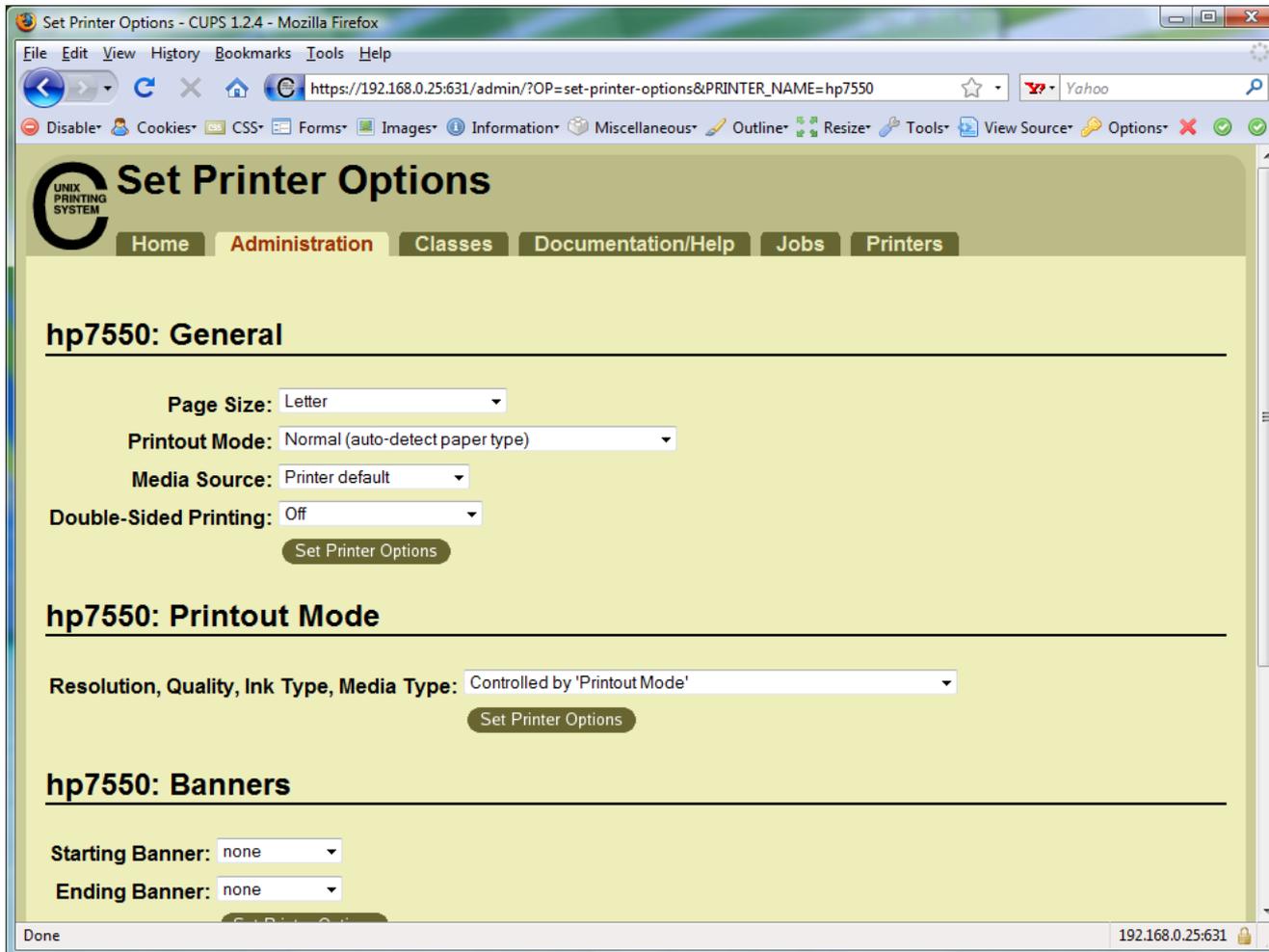
*HP PhotoSmart 7550  
Foomatic/hpijs  
(recommended) (en)*

# CUPS



*Printer has been added*

# CUPS



*View and set options as needed*

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