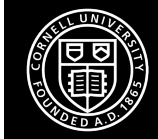
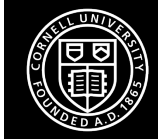


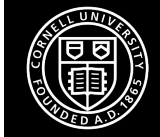
ECE 5725
Embedded Operating Systems
Lecture 7

Prof. Joseph F. Skovira



News

Lab1 Week 2



Important lab topics

Secure Shell and Secure Copy

Shell Scripts

Input/output redirection

Background

Special FS objects (FIFO)

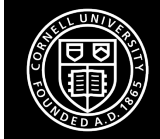
Python coding

Subprocess

GPIO and Rpi.GPIO library

piTFT documentation (schematic…)

Mplayer options



Top Ten List: Linux Commands

Roam the FS

ls
cd
pwd
mkdir
rmdir
df
tree
grep

Manipulate files

rm
cp
mv
touch
cat
head
tail
chmod
chown
‘pipe to more’

Backup

tar
gzip
dd

Cmd help

man
‘cmd’ -h

Command Commands

up/down arrow
Tab
!nn
history
ctrl-c

Processes

htop
top
ps
kill

Between hosts

ssh
scp

Find items

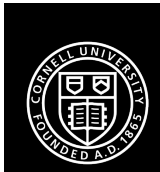
whereis
find

Control

sudo
shutdown
reboot

Network

hostname
whoami
ping
ifconfig



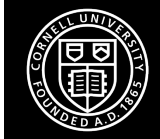
ssh and scp

ssh : secure shell

log in and execute commands on a remote machine

provide secure encrypted communication between two untrusted hosts over an insecure network

ssh jfs9@132.236.79.64



ssh and scp

scp : secure copy

provide secure encrypted file transfer between two untrusted hosts over an insecure network

Example1: logged in to the RPI, transfer files **to** ece5725-f21 server

scp –p –r /home/pi/lab1_files jfs9@132.236.79.64:/home/Lab1

Source "dot" *signon@destIP: destination folder*

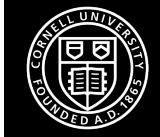
-p preserve file properties
-r recursively copy directories

Example2: logged in to the RPI, transfer files **from** ece5725-f21 server

scp –p –r my_netid@132.236.79.64:/home/jfs9/lab1_files_s21 /home/pi

Source: signon @ destIP *destination folder "dot"*

-p preserve file properties
-r recursively copy directories



Top Ten List: Linux Commands

Useful "command" commands

- up-arrow, down arrow
- tab to compete a line
- history
- history | grep ps
- !nn
- Ctrl-c

Top Ten List: Linux Commands

Processes

htop

top

ps

kill

Shell Script programming

Program to run a string of commands in the shell

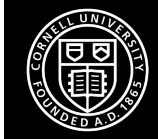
Command Line Interface
Console
Shell


```
echo ‘run ls’
ls –l
```

```
#!/bin/bash
# shell script: run ls

echo ‘run ls’
ls –l
```

File System:
/home/jfs9
shell script



Command Line Interface
Console
Shell

```
echo ‘run ls’
ls –l
```

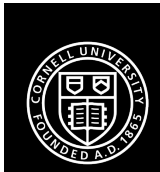
```
#!/bin/bash
# shell script: run ls

echo ‘run ls’
ls –l
```

```
echo ‘hello’ > test.txt
python t1.py &
```

File System:
/home/jfs9
shell script
test.txt

t1.py process

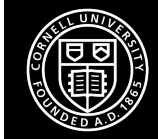


Special File Object: FIFO and Unix Domain Sockets

Pipe

FIFO

Sockets



Command Line Interface
Console
Shell

```
echo ‘run ls’
ls –l

  #!/bin/bash
  # shell script: run ls

  echo ‘run ls’
  ls –l

echo ‘hello’ > test_fifo
```

test_fifo

File System:
/home/jfs9
shell script
test_fifo

Command Line Interface
Console
Shell

```
cat test_fifo
```

Python subprocess

Allows python to execute commands in the shell

```
echo ‘hello’
echo “hello” > test_file
```

Command line

./test_script

test_script

```
#!/bin/bash
# shell script: run echo

echo “hello” > test_file
```

Shell Script

python test_code.py

test_code.py

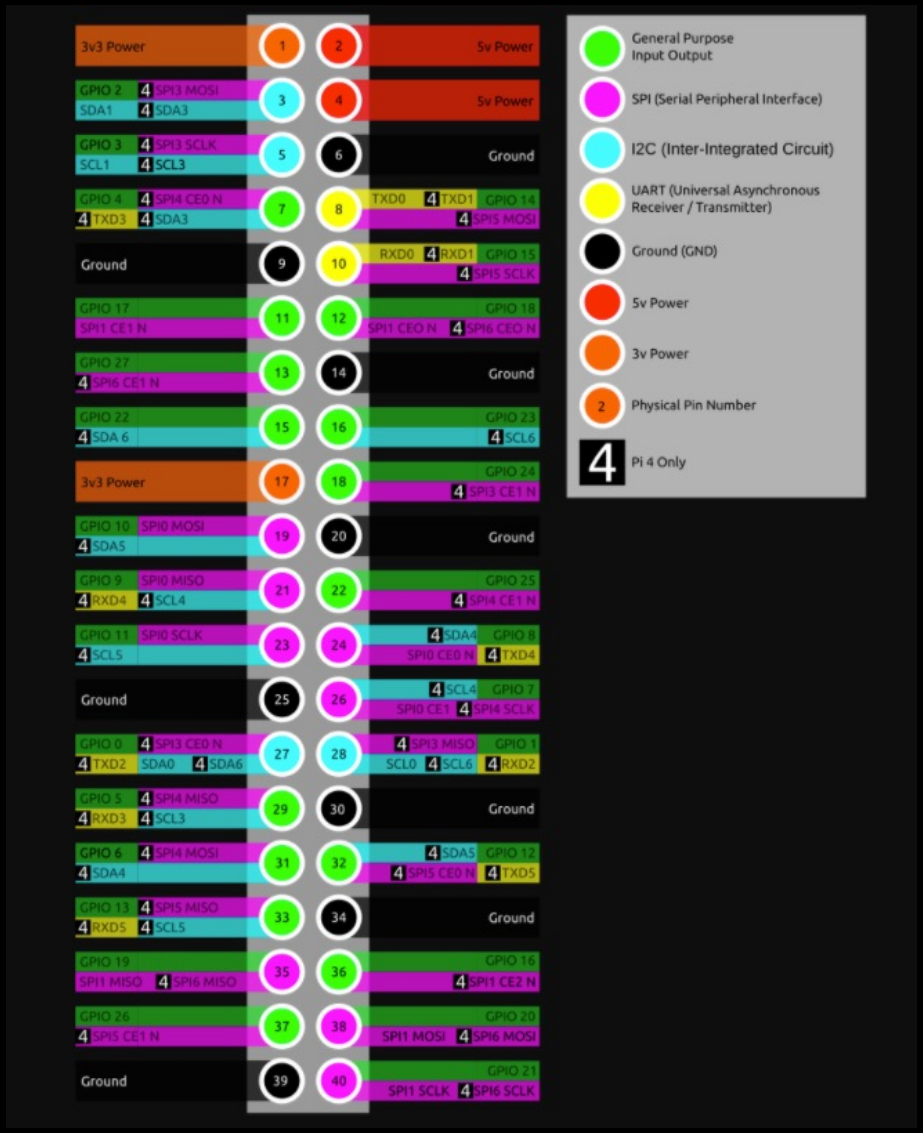
```
my_cmd =
echo “hello” > test_file

subprocess call using
my_cmd
```

Subprocess

GPIO

GPIO#	2nd func	pin#		pin#	2nd func	GPIO#
N/A	+3V3	1		2	+5V	N/A
GPIO2	SDA1 (I2C)	3		4	+5V	N/A
GPIO3	SCL1 (I2C)	5		6	GND	N/A
GPIO4	GCLK	7		8	TXD0 (UART)	GPIO14
N/A	GND	9		10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11		12	GEN1	GPIO18
GPIO27	GEN2	13		14	GND	N/A
GPIO22	GEN3	15		16	GEN4	GPIO23
N/A	+3V3	17		18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19		20	GND	N/A
GPIO9	MISO (SPI)	21		22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23		24	CE0_N (SPI)	GPIO8
N/A	GND	25		26	CE1_N (SPI)	GPIO7
(Models A and B stop here)						
EEPROM	ID_SD	27		28	ID_SC	EEPROM
GPIO5	N/A	29		30	GND	N/A
GPIO6	N/A	31		32	-	GPIO12
GPIO13	N/A	33		34	GND	N/A
GPIO19	N/A	35		36	N/A	GPIO16
GPIO26	N/A	37		38	Digital IN	GPIO20
N/A	GND	39		40	Digital OUT	GPIO21



Using Rpi.GPIO

```
import Rpi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)  # Set for broadcom numbering not board numbering
#   setup a GPIO for an input button…
#
GPIO.setup(26, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
   time.sleep(0.2)  # short sleep for screen output
   if ( not GPIO.input(26)) ):
      # Button is pressed
      print ("Button 26 has been pressed!")
```