

COMPUTER SCIENCE 501/590-003  
Computing for Science and Engineering / Life Sciences  
FALL 2009

ASSIGNMENT # 1 (maximum score is 40 points)

August 24

**Due Friday, September 11 at 1:30 pm** This assignment covers through Section 2.3 of the class notes. Read Sections 1–4, 9, and 10 of Software Carpentry.

0. (Required) Retrieve an electronic copy of this document from Blackboard Vista, and acknowledge here that you did this.
1. (4 points) Show step by step how to do the following in Unix:

- Start a long running process from the shell, in particular `find / needle >& /dev/null`
- Kill it.
- Start it again in the background.
- Bring it to the foreground.
- Suspend it and then resume it in the background.
- Start another process in the background, in particular `find / haystack >& /dev/null`.
- Run `jobs` to see that both processes are running.
- Use the command in the class notes to kill them.
- Start a new process in the background, in particular `find / needle >& /dev/null`
- Use `ps U <userid>` to find out its process id.
- Use `kill` to terminate the process.

Specifically, show for each item what you type in. Give the output for the second last item only.

2. (5 points) Assuming that `dirname` has been assigned a value using `dirname=` and that `script` has been assigned a value using `script=`, what is `bash` equivalent of the following:

```
os.mkdir(dirname)
os.rename(script, dirname + os.sep + script)
os.chdir(dirname)
open('one', 'w').close() # Hint: see below
open('two.py', 'w').close()
os.listdir(os.getcwd())
os.rename(script, os.pardir + os.sep + script)
for name in os.listdir(os.getcwd()): os.remove(name)
os.chdir(os.pardir)
os.rmdir(dirname)
```

If you want to test, you might initialize with

```
$ dirname=Test
$ touch uw.py
$ script=uw.py
```

3. (16 points) Write a Python script `ux2win.py`, which changes the name of every file in the current directory that does not contain a dot by adding the extension `.exe` to it. *However*, for any case in which there is already a file present having the extended name, this action is not taken and instead a line with the message `"name" already exists.` is printed. Also, write a second script `win2ux.py`, which strips off the extension `.exe` from every such file in the current directory unless this would conflict with an existing file name, in which case a line is printed with the message given above. Possibly useful Python functions: `os.listdir`, `os.getcwd`, `os.path.isfile`, `os.rename`. A partial tester for `ux2win.py` and `win2ux.py`:

```
import os, subprocess, sys
n = 0
while os.path.exists('Test' + str(n)): n = n + 1
dirname= 'Test' + str(n)
os.mkdir(dirname)
for script in 'ux2win.py', 'win2ux.py':
    os.rename(script, dirname + os.sep + script)
os.chdir(dirname)
open('one', 'w').close()
open('two.py', 'w').close()
open('.hidden', 'w').close()
open('three', 'w').close()
open('three.exe', 'w').close()
try:
    proc = subprocess.Popen('python ux2win.py', \
                            shell=True, stdout=subprocess.PIPE)
    stdout = proc.communicate()[0]
    if sys.platform == 'win32': stdout = stdout.replace('\r\n', '\n')
    print stdout == '"three.exe" already exists.\n'
    dirlist = os.listdir(os.getcwd()); dirlist.sort()
    print dirlist[1:5] == ['one.exe', 'three', 'three.exe', 'two.py']
    proc = subprocess.Popen('python win2ux.py', \
                            shell=True, stdout=subprocess.PIPE)
    stdout = proc.communicate()[0]
    if sys.platform == 'win32': stdout = stdout.replace('\r\n', '\n')
    print stdout == '"three" already exists.\n'
    dirlist = os.listdir(os.getcwd()); dirlist.sort()
    print dirlist[1:5] == ['one', 'three', 'three.exe', 'two.py']
except Exception, e: print e
for script in 'ux2win.py', 'win2ux.py':
    os.rename(script, os.pardir + os.sep + script)
for name in os.listdir(os.getcwd()): os.remove(name)
os.chdir(os.pardir)
os.rmdir(dirname)
```

4. (4 points) Debug the following:

```
import sys
def fib(n):
    if n == 0: return 0
    elif n == 1: return 1
    else: return fib(n-1) + fib(n-2)
```

```

for i in range(len(sys.argv)):
    arg = sys.argv[i]
    num = int(arg)
    print 'fib of', num, ' = ', fib(num)

```

5. (6 points) Write a complete Python script `echo.py`, which simply prints out its arguments. (The spacing of the output need not be identical to the spacing of the input.) For example,

```

$ python echo.py 1 2.0 three
1 2.0 three
$

```

There should be no superfluous brackets, commas, or quotes, and there should be exactly one new line.

6. (8 points) Write a Python filter that checks each line for a tab character `'\t'`. For each line that has one, output a message containing the line number, with the first line being number 1.

**Send your solutions electronically** as described below. You should send one PDF file containing your answers to problems 1, 2, and 4–6 as well as your source code for problem 3. Additionally, send your source code as a separate text file. PDF files can be created by scanning hard copy or by printing-to-PDF from a word processor or text editor or by saving-as or by using `pdflatex`. You may use plain text to represent mathematics, in which case use of simplified  $\text{\LaTeX}$  notation is recommended, e.g., caret for superscript and underscore for subscript.

1. Log on to Blackboard at <http://www.itap.purdue.edu/tlt/blackboard/> (To use Blackboard, you have to have Java enabled for your web browser.)
2. Click on Assignments from the left panel. You will find the homeworks.
3. Go into a homework assignment, you will see a submission window on the page. Below the window there is something called “select zip file”. Click the “Add Attachments” tab below it. You will be able to submit the `.zip` file containing your homework answers there.
4. Now due to the limitation of Blackboard, it accepts only `.zip` files and not other formats such as `.tar`, `.rar` or `.bz`. Therefore, you have to compress your homework answer to a `.zip` file first. There are two ways you can do it:
  - (a) Zip on your own first and when it comes to adding the attachments in Blackboard, choose “My computer” and upload your zip file directly from the computer. (Note again, it has to be a `.zip` file, otherwise even if you can upload it, you’ll not be able to submit it. That’s the requirement of Blackboard.) To pack a set of files and directories into a single file, use

```
zip <name>.zip <list>
```

and to unpack

```
unzip <name>.zip
```

- (b) Go to “My Tools” on the bottom of the left panel. Click on “My Files”, and then you will see a page where you can manage all your files for this course, and zip your file using Blackboard. If you keep all your files for this course in Blackboard, when adding the attachment for the homeworks, choose from “My Files”.
5. Submission will be closed after the due time. Please manage your time carefully.