Test 1

System Level Programming

Due July 2, 2020

Coding ( 60 points )

\*\* You will verify this with screen shots of successful commands as well as a Submission of the final

result.

PART 1 – 28 points

1. (2 points) Create a directory in the /Submissions folder in your home directory called ”Test1”. If you do not currently have a directory titled Submissions in your home directory ( ie, /Submissions ) create file

I used mkdir Test1 to create a new directory under my Submissions folder.

1. (2 points) Use the ‘find‘ command to locate the ”data.tar” fifile in my directory.

I went to cumoja1 directory to use this command to find data.tar



1. (2 points) Copy the file called ”data.tar” from my directory into your Test1 directory inside of your Submission folder

A close up of a screen

Description automatically generated

1. (2 points) Extract the contents of ”data.tar” in your Submission folder

A screenshot of a computer

Description automatically generated

1. (3 points) Of the compressed files how many lines contain the phrase ”computer science” ( ignoring case ) ?



6.(3 points) Create a file name ”computer-science” that contains all of the lines, including the line numbers, in all the files that contain the phrase ”computer science”, ignoring case?

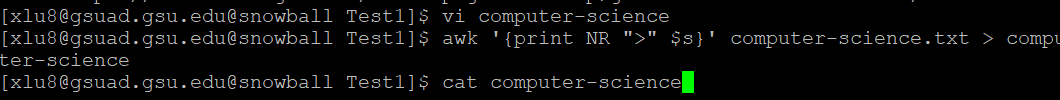
I created a new file which named computer-science.txt. and use command

Sed -n ‘/computer science/p’ data.tar > computer-science.txt to move all line which contains “computer science” in file computer-science.txt

A screen shot of a computer

Description automatically generated

Then I create another file which named “computer-science” use below command to add line number to it.



7.(3 points) Of the compressed files how many lines contain the words ”data” or ”structures” ( exact casing )?



8. (3 points) Create a file named ”data-structures” that contains all of the lines, including the line numbers, in all the files that contain the phrase words ”data” or ”structures” ( exact casing )?

 use command above to save all lines with word “data” or “structures” from “data.tar” to “data-structures.txt”



Then use command above to add line number to “data-structures”

9. (3 points) Of the compressed files how many lines contain any links to websites?



10. (3 points) Create a file named ”websites” that contains all of the lines, including the line numbers, in all the files that contain any links to websites?



Use command above to search all websites link in “data.tar” and save to “websites.txt”



Then use command above to add line number

11. (2 points) Combine the results from ”websites”, ”data-structures” and ”computer-science” using the cat command and store the results in a file called ”1.part”.



PART 2 ( Hint AWK is in a lot ) – 32 points

1. (4 points) Use a UNIX/Linux command to download the file at the following URL to your Submission

directory:

https://data.ct.gov/api/views/rybz-nyjw/rows.csv

A screenshot of a computer screen

Description automatically generated

2. (4 points) Read and format the previous document:

\*\*\*HINT It is a Comma Separated File that contains some strings denoted by quotation marks and

contains the following columns:

ID, Date, DateType, Age, Sex, Race, ResidenceCity, ResidenceCounty, ResidenceState, DeathCity,

DeathCounty, Location, LocationifOther, DescriptionofInjury, InjuryPlace, InjuryCity, InjuryCounty,

InjuryState, COD, OtherSignififican, Heroin, Cocaine, Fentanyl, FentanylAnalogue, Oxycodone, Oxymor

phone, Ethanol, Hydrocodone, Benzodiazepine, Methadone, Amphet, Tramad, Morphine NotHeroin,

Hydromorphone, Other, OpiateNOS, AnyOpioid, MannerofDeath, DeathCityGeo, ResidenceCityGeo,

InjuryCityGeo

Create a new file which called format.csv then use follow command to format the document.

**awk -F “,” ‘{print $0}’ rows.csv >> formatData.csv**

**then I use command to relace all comma with space**



3. (4 points) Parse the previous document and remove all rows the don’t have a race or sex. Save output as ”2.parse”

Create a new file which called ‘2.parse’ first. Then use follow command to remove.



A picture containing outdoor, sitting, light, water

Description automatically generated

4. (4 points) Create a new document Create a new document that only contains age, sex, race. Save output as ”2.asr”

Use command awk -F ‘,’ ‘{print ($4, $5, $6) }’ formatData >>2.asr



A screenshot of a cell phone

Description automatically generated

5. (2 points) Find the Total number of a Male and Female participants, separately.

A close up of text on a black background

Description automatically generated

A screen shot of a computer

Description automatically generated

6. (2 points) Find the average age of a Male and Female participants, separately.

A screenshot of a cell phone

Description automatically generated

A close up of a screen

Description automatically generated

7.(2 points) Find the list of the unique races that are in the study.



8. (8 points) Find the following age statistic along racial and sex lines ( ie, While Male, White Female,

Black Male, Black Female)

- Total number

- Average Age

- Min Age

- Max Age

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

9. (2 points) Save all files you used to create both parts of this and compress it all together and add it to iCollege in a tar archive.

Download them by zillaFile. Then zip them in my computer.

Comprehension – 20 points

1. (5 points) What is the following bash code doing

curl ’http://domain.com/id/[1-151468]’ -o ’#1.html’

**this command will download all pages from the website “**[**http://domain.com/id/[1-151468**](http://domain.com/id/%5b1-151468)**]” and also download 151468 files into directory as 1.html.**

grep -oh ’http://pics.domain.com/pics/original/.\*jpg’ \*.html >urls.txt

**This command will download all images from this website and save them to a file called “urls.txt”**

sort -u urls.txt | wget -i

**this command will sorts the images to download in order.**

If you would like to access in the server find ”zuckerburg.sh”

**Use Ctrl C**

2. (15 points) Detail what the code is supposed to do in each of the 5 blocks ( 3 points each ) in the ”vimeo*script.sh*”*f ilelocatedinmydirectory.*

#block 1

#

#

if [ $# -ne 1 ]; then

exit 1

Fi

ID=`echo $1 | awk -F / '{print $NF}'`

# Set the user agent ID to use

USER\_AGENT="Mozilla/5.0"

# Check we have the tools we need

which wget

if [ $? -eq 1 ]; then

echo "ERROR: this tool requires wget on the path"

exit 1 exit 1 - file not found

fi

which perl

if [ $? -eq 1 ]; then if 1 is numerically equal to 1

echo "ERROR: this tool requires perl on the path" print “ERROR: this tool requires perl on the path”

exit 1 exit 1 - file not found

Fi

**This block will check that user has passed one argument else set the value of user argument into ID variable and set USER\_AGENT variable as mozilla/5.0  
then check if we have the wget using which utility if it finds successfully . At last, find a perl tool.**

#block 2

#

#

VIDEO\_XML=`wget -U \"${USER\_AGENT}\" -q -O - http://vimeo.com/${ID}`

CONFIG\_URL=`echo $VIDEO\_XML | grep data-config-url | perl -p -e 's/^.\*? data-config-url="(.\*?)".\*$/$1/g' | perl -pe 's/&amp;/&/g'`

VIDEO\_CONFIG=`wget -U \"${USER\_AGENT}\" -q -O - ${CONFIG\_URL}`

1. **Set VIDEO\_XML as variable with the wget tool. It will download the URL using USER\_AGENT variable which get from vimeo site**
2. **CONFIG\_URL variable will export the url from XML content which is downloaded in VIDEO\_XML variable.**

**3.“VIDEO\_CONFIG” it exported URL in “CONFIG\_URL” and will download wget utility.**

#block 3

#

#

HD\_URL=`echo $VIDEO\_CONFIG | perl -pe 's/^.\*"hd":{(.\*?)}.\*$/$1/g' | perl -pe 's/^.\*"url":"(.\*?)".\*$/$1/g'`

SD\_URL=`echo $VIDEO\_CONFIG | perl -pe 's/^.\*"sd":{(.\*?)}.\*$/$1/g' | perl -pe 's/^.\*"url":"(.\*?)".\*$/$1/g'`

CAPTION=`echo $VIDEO\_XML | perl -p -e '/^.\*?\<meta property="og:title" content="(.\*?)"\>.\*$/; $\_=$1; s/[^\w.]/-/g;'`

**Find and set “HD & SD video URL” in HD\_URL and SD\_URL video using perl utility .create the caption title from the” meta property” tag**

#block 4

#

#

if [ "$HD\_URL" ]; then

DOWNLOAD\_URL=$HD\_URL

QUALITY="HD"

elif [ "$SD\_URL" ]; then

DOWNLOAD\_URL=$SD\_URL

QUALITY="SD"

else

echo "ERROR: failed to download vimeo ID ${ID}"

echo "Please report this error at https://github.com/johnteslade/vimeo-downloader/issues"

Fi

1. **Check the passed argument “HD\_URL”**
2. **Check for word “HD” in quality**
3. **check if “SD\_URL” replace it with DOWNLOAD\_URL” AND SD with “QUALITY”**
4. **check . If there is a statement print “ERROR: failed to download vimeo ID and print please report error … “ exit**

#block 5

#

#

FILENAME="${CAPTION}-(${QUALITY}-${ID}).flv"

wget -U \"${USER\_AGENT}\" -O ${FILENAME} ${DOWNLOAD\_URL}

**This will locate the file caption, quality, and ID as a .flv file .Then download the video.**

Research – 20 points

In this last port of the code you will research and detail how you would use UNIX/Linux to create the

**following tools: 1) web scrapers to pull information from website,**

**What we need first is a**[**CSS selector**](http://www.w3.org/wiki/CSS/Training/Selectors)**to locate what we are interested in. In this case it is simply a div tag with the ID “mostPopular” and you can figure this out using the Developer Tools of your favorite web browser. And now we are going to apply a chain of command line tools – each feeding their output to the next tool**

**So let’s see what is going on. First the echo pipes the URL to [wget](https://www.gnu.org/software/wget/" \o "wget" \t "_blank). I could have also provided the URL directly in the arguments but I chose to do it like this to make clear that the URL or a list of URLs itself might be the result of processing. wget fetches the HTML code from BBC, which is then normalized by [hxnormalize](http://www.w3.org/Tools/HTML-XML-utils/README" \o "HTML-XML-utilities by W3C" \t "_blank) to improve digestability by [hxselect](http://www.w3.org/Tools/HTML-XML-utils/README" \o "HTML-XML-utilities by W3C" \t "_blank) (both installed on Ubuntu by sudo apt-get install html-xml-utils), which then extracts the part of the code being identified by the CSS selector.**[**Lynx**](http://lynx.isc.org/)**finally turns the code into a layouted text that you would see in a browser.**

**Cite:**

# “Using the Linux Shell for Web Scraping” 28 Feb. 2014, <https://www.joyofdata.de/blog/using-linux-shell-web-scraping/>

**2) email spam software that pulls emails address from websites and sends them spam emails ,**

**To download email spam software that extracts email addresses from websites and sends them spam, this is called spoofing. The first thing you need to do is to install a supported operating system. For example, you can use a relay server, which is an SMTP server trusted by major companies, as an authorized sender of email. Create a Mailborder account on this site, then verify your account and log in. Setting DNS requirements, running a DNS server may have many benefits. First of all, it is interesting and educational. Secondly, you are a control freak and hope you have as many websites as possible (Pstatz). Run the repository installation script and then install through the system package manager**

**Cite:**

Pstatz. “Set Up a DNS Name Server.” Wired, Conde Nast, 15 Feb. 2010, www.wired.com/2010/02/set\_up\_a\_dns\_name\_server/.

**3) bots that grab information from live sites to make business decisions.**

Bots are made by different methods. This can be achieved by writing a script and a botnet to create an email address, another bot to create a Twitter account and using these addresses, and sometimes writing a third botnet to create a phone number to verify those social media accounts .