**CS3377 Assignment 2 Due: 7/06 Monday Noon.**

Last Name: \_\_\_Pulliam\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

First Name: \_\_\_\_Drew\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NetID (email): \_\_\_DTP180003@utdallas.edu\_\_\_\_\_\_\_\_

Submission requirements.

Submit (1) a word document (this file with your answers and listing of any program & copy and paste its run-log in a professional and presentable format and style) and (2) a zip file of a folder which will contain all the codes (all the source codes and executables, Makefile, and the instruction for how to compile and run) and a document file (.doc or .docx).

This word document (this file) is your documentation (as a basis to add your answers) to be included here - all your answers, all your program-listings, instructions to compile and run [screen-shots, terminal text, or session log] to show your work done, including how to compile and run for each cases.

The word file should also have (1) in header with the course & section number, your name (Name: last name, first name), your netID (email), and Assignment #, (2) in footer with page number, (3) line number (restart from each page in left margin. Page Layout => Line Numbers). (4) Your program should have some comments (minimal or reasonable), to tell the code-reader what a segment of code is doing, and with the comment in the front/head of the program about you, this course and assignment, etc. (you may copy and paste some out of this document for your comment).

\*\* Your executable codes (that you submit) should run in cs1.utdallas.edu without any change or recompilation.

\*\* Bring this cover sheet (this page) to TA for your demo. (Please check with TA for demo or schedule).

\*\* Upload this document file (with your answers) and a zip file (containing all the codes [source and binary etc.].

**Scoresheet**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Parts** |  |  |  |  |  |  |  |
| **Part1**  **60%** | **Week04 Lab1 find** | **Week04 Lab2 grep** | **Week04**  **Lab3 vi/vim** |  |  |  |  |
|  | **Week05**  **Lab1 Sobell ch08 bash** | **Week05**  **Lab2 Sobell ch10 bash** |  |  |  |  |  |
|  | **Week06**  **Lab1 APUE code install** | **Week06**  **Lab2 APUE**  **Ch01 codes** | **Week06**  **Lab3**  **Zip tar gzip** |  |  |  |  |
| **Part2 40%**  Commands  & File, File System in C/C++  **CLO1,2,4,8** | **#1** | **#2** | **#3** | **#4** |  |  | **Demo &**  **Document** |
| **Total** |  |  |  |  |  |  |  |

Note: CLO means Student Course Learning Objective (as listed in Syllabus)

**Student Learning Objectives/Outcomes**

1. Ability to use the UNIX operating system interactively as a user (commands)

2. Ability to express algorithmic solutions using shell scripting (utilities)

3. Ability to understand and use regular expressions

4. Ability to use the UNIX programming environment (editor, compiler and linker)

5. Ability to understand UNIX processes (creation and control)

6. Ability to perform input/output of binary files

7. Ability to use interprocess communication (pipes, sockets and signals)

8. Ability to understand the UNIX file system

9. Ability to understand and use version control system

|  |
| --- |
| Deduction - Documentation (this .doc file) and upload  Max -10% if not done or poorly prepared |
| Deduction for Demo (Demo time-slots may be scheduled by TA later, for you to do the demo).  Max -10%. |

Note. Any "poor" documentation (that is, this document with your answers etc.) may result in a penalty (up to -10%).

Upload this document file (this file with your answers) and a zip file (containing all the codes [source and binary etc.] and its run log or results. All the code should run in cs1 without any change).

\*\* A demo (of your assignment) will be announced and scheduled by TA. Your demo should be done within the week of the due date. For any scheduling conflict, please contact or consult with TA (or the instructor) for your situation, and/or for alternate time or discretion as soon as possible, before the due.

**Assignment 2 Part1**

Part1 consists of weekly activity items (labs) in Week04, Week05, and Week06 Activity folders:

Week04 Lab1 – find

Week04 Lab2 – grep

Week04 Lab3 – vi/vim editor

Week05 Lab1 – Sobell ch08 bash

Week05 Lab2 – Sobell Ch10 bash

Week06 Lab1 – APUE Code install

Week06 Lab2 – APUE Ch01 Code try & run

Week06 Lab3 – File Archive and compression with zip, tar, gzip

Provide each of your lab (that you have prepared and submitted) in your lab document below (copy and paste below).

Have a page break for each lab to start. Provide a proper heading and subheading of each lab (and each part of the lab) in a presentable manner. You may view and take the example of the layout and format from the sample lab report.

For the log or output of your session (e.g., in cs1), you should use the font (Courier New) to have an alignment of the text and the font size (8 to 10) as shown below.

{cslinux1:~} cat hello.c

#include <stdio.h>

#include <stdlib.h>

int main()

{

// this is richard min

printf("Hello World\n"); // OK waw hello Richard and ok Richard

exit(0); // end of the program

}

{cslinux1:~} gcc hello.c -o hello

{cslinux1:~} ./hello

Hello World

Place your lab reports in the next page and thereafter.

**Your Lab reports to be placed here.**

**CS3377.0W1 Week 4 Lab 1**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Test find for files modified more than 900 days ago**

{cslinux1:~/cs3377/w4} find /usr/bin/ -mtime +900

/usr/bin/docbook2html

/usr/bin/ldns-read-zone

...

{cslinux1:~/cs3377/w4} find /usr/bin/ -mtime +1200 | wc -l

898

1. **Find files that begin with ‘c’**

{cslinux1:~/cs3377/w4} find /usr/bin/ -name 'c\*'

/usr/bin/cat

/usr/bin/chem

...

1. **Find files smaller than 100 bytes**

{cslinux1:~/cs3377/w4} find /usr/bin/ -size -100

/usr/bin/docbook2html

/usr/bin/jarsigner

/usr/bin/dvi2fax

1. **Only list the first 10 files in /usr**

{cslinux1:~/cs3377/w4} find /usr/bin/ | head

/usr/bin/

/usr/bin/docbook2html

/usr/bin/jarsigner

/usr/bin/dvi2fax

/usr/bin/ldns-read-zone

/usr/bin/dvipdf

/usr/bin/plymouth

/usr/bin/abrt-action-generate-backtrace

/usr/bin/lsmcli

/usr/bin/gnomevfs-df

1. **Find anything that contains ‘sh’ and is a symbolic link**

{cslinux1:~/cs3377/w4} find /usr/bin/ -type l -name '\*sh\*'

/usr/bin/mshortname

/usr/bin/mshowfat

/usr/bin/sh

/usr/bin/csh

/usr/bin/bashbug

/usr/bin/fig2ps2tex.sh

/usr/bin/texhash

/usr/bin/mate-panel-screenshot

/usr/bin/wish

/usr/bin/tclsh

* 1. **Use -exec to do ls -l on the same files as step 5 above**

{cslinux1:~/cs3377/w4} find /usr/bin/ -type l -name '\*sh\*' -exec ls -l {} \;

lrwxrwxrwx. 1 root root 6 Apr 6 2016 /usr/bin/mshortname -> mtools

lrwxrwxrwx. 1 root root 6 Apr 6 2016 /usr/bin/mshowfat -> mtools

lrwxrwxrwx 1 root root 4 May 20 10:52 /usr/bin/sh -> bash

lrwxrwxrwx 1 root root 4 May 20 11:03 /usr/bin/csh -> tcsh

lrwxrwxrwx 1 root root 10 May 20 10:52 /usr/bin/bashbug -> bashbug-64

lrwxrwxrwx 1 root root 10 May 3 2016 /usr/bin/fig2ps2tex.sh -> fig2ps2tex

lrwxrwxrwx 1 root root 8 May 20 10:52 /usr/bin/texhash -> mktexlsr

lrwxrwxrwx 1 root root 15 Jan 9 2019 /usr/bin/mate-panel-screenshot -> mate-screenshot

lrwxrwxrwx. 1 root root 7 Apr 6 2016 /usr/bin/wish -> wish8.5

lrwxrwxrwx. 1 root root 8 Apr 6 2016 /usr/bin/tclsh -> tclsh8.5

**CS3377.0W1 Week 4 Lab 2**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Use grep to display lines with ‘model’**

{cslinux1:~} grep model /proc/cpuinfo

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

model : 45

model name : Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz

1. **Display number of lines with ‘Remote’**

{cslinux1:~} grep Remote /etc/services -c

202

1. **Display number of lines with string (and word) ‘send’**

{cslinux1:~} grep send /etc/services -c

17

{cslinux1:~} grep send /etc/services -c -w

13

1. **Display number of lines with ‘send’ ignoring case**

{cslinux1:~} grep -i send /etc/services -c

19

1. **Display number of lines without ‘send’**

{cslinux1:~} grep -i send /etc/services -v -c

11157

1. **Display lines containing ‘send’ and their corresponding line numbers**

{cslinux1:~} grep -i send /etc/services -n

37:msp 18/tcp # message send protocol (historic)

38:msp 18/udp # message send protocol (historic)

609:mpm-snd 46/tcp # MPM [default send]

610:mpm-snd 46/udp # MPM [default send]

763:send 169/tcp # SEND

764:send 169/udp # SEND

963:srssend 362/tcp # SRS Send

964:srssend 362/udp # SRS Send

1368:sift-uft 608/tcp # Sender-Initiated/Unsolicited File Transfer

1369:sift-uft 608/udp # Sender-Initiated/Unsolicited File Transfer

1567:netviewdm2 730/tcp # IBM NetView DM/6000 send/tcp

1568:netviewdm2 730/udp # IBM NetView DM/6000 send/tcp

1584:tell 754/udp # send

1657:ideafarm-door 902/udp # self documenting Door: send 0x00 for info

1659:ideafarm-panic 903/udp # self documenting Panic Door: send 0x00 for info

3558:#ipsendmsg 1992/tcp # IPsendmsg

3559:#ipsendmsg 1992/udp # IPsendmsg

9179:skip-cert-send 6456/tcp # SKIP Certificate Send

9180:skip-cert-send 6456/udp # SKIP Certificate Send

1. **Display 10 lines (head) containing word ‘27’**

{cslinux1:~} grep -r 27 /usr/share/ -w | head

Binary file /usr/share/plymouth/themes/charge/background-tile.png matches

/usr/share/gtk-doc/html/libbonobo/libbonobo-bonobo-persist-file.html:<a name="id3180905"></a><p class="title"><b>Example 27. Chaining to a <span class="type">PersistStream</span> implementation</b></p>

/usr/share/gtk-doc/html/libbonobo/libbonobo-bonobo-persist-file.html:27

/usr/share/gtk-doc/html/libbonobo/libbonobo-faq.html:<a name="id3188416"></a><a name="id3188418"></a><p><b>27.</b></p>

Binary file /usr/share/gtk-doc/html/libbonobo/running-context.png matches

/usr/share/gtk-doc/html/libbonobo/libbonobo-bonobo-persist-stream.html:27

/usr/share/gtk-doc/html/libbonobo/libbonobo-bonobo-property-bag.html:27

/usr/share/gtk-doc/html/harfbuzz/plans-and-caching.html:<meta name="generator" content="GTK-Doc V1.27.1 (XML mode)">

/usr/share/gtk-doc/html/harfbuzz/plans-and-caching.html:<hr>Generated by GTK-Doc V1.27.1</div>

/usr/share/gtk-doc/html/harfbuzz/harfbuzz-hb-uniscribe.html:<meta name="generator" content="GTK-Doc V1.27.1 (XML mode)">

1. **Display number of times word ‘27’ occurs in files at least once (all files minus files with count of 0)**

{cslinux1:~} grep -rcw 27 /usr/share/ | grep -vc 0

grep: /usr/share/polkit-1/rules.d: Permission denied

14686

{cslinux1:~}

**CS3377.0W1 Week 4 Lab 3**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

{cslinux1:~/cs3377/w4} vim pizza

{cslinux1:~/cs3377/w4} cat pizza

PIZZA

Pizza is an oven-baked, flat, round bread

typically covered with a tomatoe sauce, cheese

and various toppings. Pizza was

invented in Naples, and the dish has

since become popular in many parts of the world.

(from Wikipedia)

**CS3377.0W1 Week 5 Lab 1 PART 1**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Create short**

{cslinux1:~/cs3377} vim short

{cslinux1:~/cs3377} ls -l

total 280

drwx--x--x 5 dtp180003 sn 186 Jun 15 16:44 A1-dtp180003

-rw------- 1 dtp180003 sn 889 Jun 15 16:43 A1-dtp180003.gzip

-rw------- 1 dtp180003 sn 20480 Jun 15 16:39 A1-dtp180003.tar

-rw------- 1 dtp180003 sn 3412 Jun 15 16:53 A1-dtp180003.zip

-rw------- 1 dtp180003 sn 17 Jun 27 17:17 short

drwx--x--x 2 dtp180003 sn 23 Jun 20 17:49 w4

1. **Try and execute**

{cslinux1:~/cs3377} ./short

-bash: ./short: Permission denied

1. **Use chmod to make executable**

{cslinux1:~/cs3377} chmod 700 short

{cslinux1:~/cs3377} ls -l

total 304

drwx--x--x 5 dtp180003 sn 186 Jun 15 16:44 A1-dtp180003

-rw------- 1 dtp180003 sn 889 Jun 15 16:43 A1-dtp180003.gzip

-rw------- 1 dtp180003 sn 20480 Jun 15 16:39 A1-dtp180003.tar

-rw------- 1 dtp180003 sn 3412 Jun 15 16:53 A1-dtp180003.zip

-rwx------ 1 dtp180003 sn 17 Jun 27 17:17 short

drwx--x--x 2 dtp180003 sn 23 Jun 20 17:49 w4

{cslinux1:~/cs3377} ./short

hi there

1. **Add comment and #!/bin/bash**

{cslinux1:~/cs3377} vim short

{cslinux1:~/cs3377} cat short

#!/bin/bash

echo 'hi there'

# this script prints "hi there"

{cslinux1:~/cs3377} ./short

hi there

1. **Write first to display first argument**

{cslinux1:~/cs3377} vim first

{cslinux1:~/cs3377} chmod 700 first

{cslinux1:~/cs3377} ./first -q

first arg = -q

{cslinux1:~/cs3377} ./first banana

first arg = banana

{cslinux1:~/cs3377} cat first

#!/bin/bash

echo "first arg = $1"

# this script prints the first argument it is called with

1. **Write a script to copy a file to “file”.bak**

{cslinux1:~/cs3377} vim copyScript

{cslinux1:~/cs3377} cat copyScript

#!/bin/bash

cp $1 $1.bak

{cslinux1:~/cs3377} chmod 700 copyScript

{cslinux1:~/cs3377} ./copyScript first

{cslinux1:~/cs3377} ls

A1-dtp180003 A1-dtp180003.gzip A1-dtp180003.tar A1-dtp180003.zip copyScript first first.bak short w4

{cslinux1:~/cs3377} cat first

#!/bin/bash

echo "first arg = $1"

# this script prints the first argument it is called with

{cslinux1:~/cs3377} cat first.bak

#!/bin/bash

echo "first arg = $1"

# this script prints the first argument it is called with

1. **Create a file with a space in the name**

{cslinux1:~/cs3377} touch 'two words'

{cslinux1:~/cs3377} ls

A1-dtp180003 A1-dtp180003.gzip A1-dtp180003.tar A1-dtp180003.zip copyScript first first.bak short two words w4

{cslinux1:~/cs3377} ./copyScript 'two words'

cp: target ‘words.bak’ is not a directory

{cslinux1:~/cs3377} vim copyScript

{cslinux1:~/cs3377} ./copyScript 'two words'

{cslinux1:~/cs3377} ls

A1-dtp180003 A1-dtp180003.tar copyScript first.bak two words w4

A1-dtp180003.gzip A1-dtp180003.zip first short two words.bak

{cslinux1:~/cs3377} cat copyScript

#!/bin/bash

cp "$1" "$1".bak

**CS3377.0W1 Week 5 Lab 1 PART 2**

1. **Create and test myname variable**

{cslinux1:~/cs3377} myname=drew

{cslinux1:~/cs3377} echo $myname

drew

{cslinux1:~/cs3377} echo "$myname"

drew

{cslinux1:~/cs3377} echo '$myname'

$myname

1. **Make myname read only**

{cslinux1:~/cs3377} readonly myname

{cslinux1:~/cs3377} myname=john

-bash: myname: readonly variable

1. **What is value of HOME and ~?**

{cslinux1:~/cs3377} echo $HOME

/home/013/d/dt/dtp180003

{cslinux1:~/cs3377} echo ~

/home/013/d/dt/dtp180003

1. **What is current PATH? Append ~/bin**

{cslinux1:~/cs3377} echo $PATH

/usr/local/bin:/usr/bin:/bin:/usr/ccs/bin:/sbin:/usr/sbin:/usr/local/openwin/bin:/usr/openwin/bin

{cslinux1:~/cs3377} PATH=$PATH:~/bin

{cslinux1:~/cs3377} echo $PATH

/usr/local/bin:/usr/bin:/bin:/usr/ccs/bin:/sbin:/usr/sbin:/usr/local/openwin/bin:/usr/openwin/bin:/home/013/d/dt/dtp180003/bin

1. **Change PS1 to ‘$’ / ’#’**

{cslinux1:~/cs3377} echo $PS1

{\h:\w}

{cslinux1:~/cs3377} PS1='\$ '

1. **Write script that shows date + PATH**

$ date

Sat Jun 27 18:29:39 CDT 2020

$ vim infoScript

$ chmod 700 infoScript

$ ./infoScript

date = Sat Jun 27 18:33:46 CDT 2020

HOME = /home/013/d/dt/dtp180003

PATH = /usr/local/bin:/usr/bin:/bin:/usr/ccs/bin:/sbin:/usr/sbin:/usr/local/openwin/bin:/usr/openwin/bin:/home/013/d/dt/dtp180003/bin

**CS3377.0W1 Week 5 Lab 1 PART 3**

1. **Type the following commands (output not shown to save space)**

$ head /etc/services

...

$ ls /etc/

...

$ who

...

$ date

Sat Jun 27 18:35:22 CDT 2020

$ uname -a

Linux cslinux1.utdallas.edu 3.10.0-1127.8.2.el7.x86\_64 #1 SMP Tue May 12 16:57:42 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux

1. **Show 10 most recent commands**

$ history | tail

109 ./infoScript

110 vim infoScript

111 ./infoScript

112 LAB3

113 head /etc/services

114 ls /etc/

115 who

116 date

117 uname -a

118 history | tail

1. **Use ! to reference previous events**

$ !114

ls /etc/

$ history | tail

111 ./infoScript

112 LAB3

113 head /etc/services

114 ls /etc/

115 who

116 date

117 uname -a

118 history | tail

119 ls /etc/

120 history | tail

$ !-4

uname -a

Linux cslinux1.utdallas.edu 3.10.0-1127.8.2.el7.x86\_64 #1 SMP Tue May 12 16:57:42 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux

$ !!

uname -a

Linux cslinux1.utdallas.edu 3.10.0-1127.8.2.el7.x86\_64 #1 SMP Tue May 12 16:57:42 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux

$ !head

head /etc/services

# /etc/services:

# $Id: services,v 1.55 2013/04/14 ovasik Exp $

#

# Network services, Internet style

# IANA services version: last updated 2013-04-10

#

# Note that it is presently the policy of IANA to assign a single well-known

# port number for both TCP and UDP; hence, most entries here have two entries

# even if the protocol doesn't support UDP operations.

# Updated from RFC 1700, ``Assigned Numbers'' (October 1994). Not all ports

1. **Create alias d = date**

$ alias d='date'

$ d

Sat Jun 27 21:47:33 CDT 2020

$ alias d

alias d='date'

1. **Create and remove alias ls = ls -l**

$ alias ls='ls -l'

$ ls

total 544

drwx--x--x 5 dtp180003 sn 186 Jun 15 16:44 A1-dtp180003

-rw------- 1 dtp180003 sn 889 Jun 15 16:43 A1-dtp180003.gzip

-rw------- 1 dtp180003 sn 20480 Jun 15 16:39 A1-dtp180003.tar

-rw------- 1 dtp180003 sn 3412 Jun 15 16:53 A1-dtp180003.zip

-rwx------ 1 dtp180003 sn 29 Jun 27 17:52 copyScript

-rwx------ 1 dtp180003 sn 92 Jun 27 17:28 first

-rwx------ 1 dtp180003 sn 92 Jun 27 17:34 first.bak

-rwx------ 1 dtp180003 sn 74 Jun 27 18:33 infoScript

-rwx------ 1 dtp180003 sn 60 Jun 27 17:26 short

-rw------- 1 dtp180003 sn 0 Jun 27 17:50 two words

-rw------- 1 dtp180003 sn 0 Jun 27 17:52 two words.bak

drwx--x--x 2 dtp180003 sn 23 Jun 20 17:49 w4

$ 'ls'

A1-dtp180003 A1-dtp180003.tar copyScript first.bak short two words.bak

A1-dtp180003.gzip A1-dtp180003.zip first infoScript two words w4

$ unalias ls

$ ls

A1-dtp180003 A1-dtp180003.tar copyScript first.bak short two words.bak

A1-dtp180003.gzip A1-dtp180003.zip first infoScript two words w4

1. **Write one-line welcome function**

$ welcome() { echo "Welcome to my computer." ; }

$ welcome

Welcome to my computer.

1. **Write long function that does date + ls -l**

$ function long () {

> date

> ls -l

> }

$ long

Sat Jun 27 21:51:46 CDT 2020

total 544

drwx--x--x 5 dtp180003 sn 186 Jun 15 16:44 A1-dtp180003

-rw------- 1 dtp180003 sn 889 Jun 15 16:43 A1-dtp180003.gzip

-rw------- 1 dtp180003 sn 20480 Jun 15 16:39 A1-dtp180003.tar

-rw------- 1 dtp180003 sn 3412 Jun 15 16:53 A1-dtp180003.zip

-rwx------ 1 dtp180003 sn 29 Jun 27 17:52 copyScript

-rwx------ 1 dtp180003 sn 92 Jun 27 17:28 first

-rwx------ 1 dtp180003 sn 92 Jun 27 17:34 first.bak

-rwx------ 1 dtp180003 sn 74 Jun 27 18:33 infoScript

-rwx------ 1 dtp180003 sn 60 Jun 27 17:26 short

-rw------- 1 dtp180003 sn 0 Jun 27 17:50 two words

-rw------- 1 dtp180003 sn 0 Jun 27 17:52 two words.bak

drwx--x--x 2 dtp180003 sn 23 Jun 20 17:49 w4

**CS3377.0W1 Week 5 Lab 2**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Create a script named all that displays calling program + parameters**

{cslinux1:~/cs3377} vim all

{cslinux1:~/cs3377} chmod 700 all

{cslinux1:~/cs3377} ./all

calling program = ./all

number of parameters = 0

parameters =

{cslinux1:~/cs3377} ./all one

calling program = ./all

number of parameters = 1

parameters = one

{cslinux1:~/cs3377} ./all one two three four five

calling program = ./all

number of parameters = 5

parameters = one two three four five

1. **Create symbolic link linkto for all script**

{cslinux1:~/cs3377} ln -s all linkto

{cslinux1:~/cs3377} ./linkto

calling program = ./linkto

number of parameters = 0

parameters =

1. **Create script myname that prompts user for their name**

{cslinux1:~/cs3377} vim myname

{cslinux1:~/cs3377} chmod 700 myname

{cslinux1:~/cs3377} ./myname

Enter your name:

Drew

Hello Drew

1. **Rewrite myname2 to print to a file instead of the screen**

{cslinux1:~/cs3377} vim myname2

{cslinux1:~/cs3377} chmod 700 myname2

{cslinux1:~/cs3377} ./myname2

Enter your name:

john

wrote to 35086.name

{cslinux1:~/cs3377} cat 35086.name

John

1. **Write script looper that prints all arguments on separate lines**

{cslinux1:~/cs3377} vim looper

{cslinux1:~/cs3377} chmod 700 looper

{cslinux1:~/cs3377} ./looper one two three

one

two

three

{cslinux1:~/cs3377} cat looper

#!/bin/bash

for arg

do

echo "$arg"

done

1. **Rewrite looper 2 to do the same with a for…in control structure**

{cslinux1:~/cs3377} vim looper2

{cslinux1:~/cs3377} chmod 700 looper2

{cslinux1:~/cs3377} ./looper2 one two three

one

two

three

{cslinux1:~/cs3377} cat looper2

#!/bin/bash

for i in $@

do

echo "$i"

done

1. **Write script ifthen that prompts user and checks for null input**

{cslinux1:~/cs3377} vim ifthen

{cslinux1:~/cs3377} chmod 700 ifthen

{cslinux1:~/cs3377} ./ifthen

>>

Where is your input?

{cslinux1:~/cs3377} ./ifthen

>> I'm hungry

You entered: I'm hungry

{cslinux1:~/cs3377} cat ifthen

#!/bin/bash

read -p ">> " userInput

if [[ -z "$userInput" ]]; then

echo "Where is your input?"

exit 1

else

echo "You entered: $userInput"

fi

**CS3377.0W1 Week 6 Lab 1**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Uncompress gzip file**

{cslinux1:~/cs3377} gunzip src.3e.tar.gz

**2. Unzip tar file (output trimmed)**

{cslinux1:~/cs3377} tar -xvf src.3e.tar

apue.3e/

apue.3e/advio/

apue.3e/daemons/

...

**3. Cd to apue.3e**

{cslinux1:~/cs3377} cd apue.3e/

**4. Make executables (output trimmed)**

{cslinux1:~/cs3377/apue.3e} make

for i in lib intro sockets advio daemons datafiles db environ fileio filedir ipc1 ipc2 proc pty relation signals standards stdio termios threadctl threads printer exercises; do \

(cd $i && echo "making $i" && make ) || exit 1; \

done

making lib

make[1]: Entering directory `/home/013/d/dt/dtp180003/cs3377/apue.3e/lib'

gcc -ansi -I../include -Wall -DLINUX -D\_GNU\_SOURCE -c -o bufargs.o bufargs.c

gcc -ansi -I../include -Wall -DLINUX -D\_GNU\_SOURCE -c -o cliconn.o cliconn.c

gcc -ansi -I../include -Wall -DLINUX -D\_GNU\_SOURCE -c -o clrfl.o clrfl.c

...

**6. Cd intro**

{cslinux1:~/cs3377/apue.3e} cd intro/

**7. Cat hello.c to see contents**

{cslinux1:~/cs3377/apue.3e/intro} cat hello.c

#include "apue.h"

int

main(void)

{

printf("hello world from process ID %ld\n", (long)getpid());

exit(0);

}

**8. run hello.c**

{cslinux1:~/cs3377/apue.3e/intro} ./hello

hello world from process ID 3956

**9. run shell1.c**

{cslinux1:~/cs3377/apue.3e/intro} ./shell1

% ls

getcputc hello ls1 Makefile mycat.c shell1.c shell2.c testerror.c uidgid.c

getcputc.c hello.c ls1.c mycat shell1 shell2 testerror uidgid

% ^C

{cslinux1:~/cs3377/apue.3e/intro}

**CS3377.0W1 Week 6 Lab 2**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

1. **Create apue01 directory**

{cslinux1:~/cs3377} mkdir apue01

{cslinux1:~/cs3377} cd apue01/

1. **Copy chapter 1 sample programs to this directory**

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.10" fig1.10.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.3" fig1.3.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.4" fig1.4.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.5" fig1.5.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.6" fig1.6.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.7" fig1.7.c

{cslinux1:~/cs3377/apue01} cp

"/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.9" fig1.9.c

{cslinux1:~/cs3377/apue01} cp "/home/013/d/dt/dtp180003/cs3377/apue.3e/figlinks/fig1.8" fig1.8.c

{cslinux1:~/cs3377/apue01} ls

fig1.10.c fig1.3.c fig1.4.c fig1.5.c fig1.6.c fig1.7.c fig1.8.c fig1.9.c

1. **Compile all chapter 1 sample programs to fig1.x.exe**

{cslinux1:~/cs3377/apue01} gcc fig1.3.c -o fig1.3.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.4.c -o fig1.4.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.5.c -o fig1.5.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.6.c -o fig1.6.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.7.c -o fig1.7.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.8.c -o fig1.8.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.9.c -o fig1.9.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} gcc fig1.10.c -o fig1.10.exe -I ~/cs3377/apue.3e/include/ -L ~/cs3377/apue.3e/lib/ -lapue

{cslinux1:~/cs3377/apue01} ls

fig1.10.c fig1.3.c fig1.4.c fig1.5.c fig1.6.c fig1.7.c fig1.8.c fig1.9.c

fig1.10.exe fig1.3.exe fig1.4.exe fig1.5.exe fig1.6.exe fig1.7.exe fig1.8.exe fig1.9.exe

1. **Run fig1.3 (basically ls) on current directory**

{cslinux1:~/cs3377/apue01} ./fig1.3.exe ./

.

..

fig1.10.exe

fig1.7.c

fig1.6.c

fig1.10.c

fig1.4.exe

fig1.9.exe

fig1.3.c

fig1.8.c

fig1.9.c

fig1.4.c

fig1.6.exe

fig1.3.exe

fig1.5.c

fig1.7.exe

fig1.5.exe

fig1.8.exe

1. **Use fig1.4 to write standard input to standard output (in this case write terminal to file named output.txt)**

{cslinux1:~/cs3377/apue01} ./fig1.4.exe > output.txt

hello this is my 1.4 test

^C

{cslinux1:~/cs3377/apue01} cat output.txt

hello this is my 1.4 test

1. **Use fig1.5 to copy output.txt from fig1.4 to a new file named output2.txt**

{cslinux1:~/cs3377/apue01} ./fig1.5.exe < output.txt > output2.txt

{cslinux1:~/cs3377/apue01} cat output2.txt

hello this is my 1.4 test

1. **Execute fig1.6 (hello world + process ID)**

{cslinux1:~/cs3377/apue01} ./fig1.6.exe

hello world from process ID 31512

1. **Execute fig1.7 (which is basically a shell) run ls to check that it works then exit**

{cslinux1:~/cs3377/apue01} ./fig1.7.exe

% ls

fig1.10.c fig1.3.c fig1.4.c fig1.5.c fig1.6.c fig1.7.c fig1.8.c fig1.9.c output2.txt

fig1.10.exe fig1.3.exe fig1.4.exe fig1.5.exe fig1.6.exe fig1.7.exe fig1.8.exe fig1.9.exe output.txt

% ^C

1. **Execute fig1.8 (basically writes out errors for you)**

{cslinux1:~/cs3377/apue01} ./fig1.8.exe

EACCES: Permission denied

./fig1.8.exe: No such file or directory

1. **Execute fig1.9 (print user ID and group ID)**

{cslinux1:~/cs3377/apue01} ./fig1.9.exe

uid = 633083, gid = 215

1. **Execute fig1.10 (also a shell like 1.7, but it gives an interrupt message when finished)**

{cslinux1:~/cs3377/apue01} ./fig1.10.exe

% ls

fig1.10.c fig1.3.c fig1.4.c fig1.5.c fig1.6.c fig1.7.c fig1.8.c fig1.9.c output2.txt

fig1.10.exe fig1.3.exe fig1.4.exe fig1.5.exe fig1.6.exe fig1.7.exe fig1.8.exe fig1.9.exe output.txt

% ^Cinterrupt

**CS3377.0W1 Week 6 Lab 3**

**Drew Pulliam - DTP180003**

**(Documentation with Terminal Log)**

┌────────────────────────────────────────────────────────────────────┐

│ • MobaXterm 20.2 • │

│ (SSH client, X-server and networking tools) │

│ │

│ → SSH session to dtp180003@cslinux1.utdallas.edu │

│ • SSH compression : v │

│ • SSH-browser : v │

│ • X11-forwarding : v (remote display is forwarded through SSH) │

│ • DISPLAY : v (automatically set on remote server) │

│ │

│ → For more info, ctrl+click on help or visit our website │

└────────────────────────────────────────────────────────────────────┘

**0. Create archive directory**

{cslinux1:~/cs3377} mkdir archive

{cslinux1:~/cs3377} cd archive/

**1. zip hello and hello.c, then delete the files and unzip them**

{cslinux1:~/cs3377/archive} ls

hello hello.c

{cslinux1:~/cs3377/archive} zip hello.zip hello hello.c

adding: hello (deflated 72%)

adding: hello.c (deflated 4%)

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.zip

{cslinux1:~/cs3377/archive} rm hello

rm: remove regular file ‘hello’? y

{cslinux1:~/cs3377/archive} rm hello.c

rm: remove regular file ‘hello.c’? y

{cslinux1:~/cs3377/archive} ls

hello.zip

{cslinux1:~/cs3377/archive} unzip hello.zip

Archive: hello.zip

inflating: hello

inflating: hello.c

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.zip

**2. Create a tar with hello and hello.c, then delete the files and restore them**

{cslinux1:~/cs3377/archive} tar -cvf hello.tar hello hello.c

hello

hello.c

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.tar hello.zip

{cslinux1:~/cs3377/archive} rm hello

rm: remove regular file ‘hello’? y

{cslinux1:~/cs3377/archive} rm hello.c

rm: remove regular file ‘hello.c’? y

{cslinux1:~/cs3377/archive} ls

hello.tar hello.zip

{cslinux1:~/cs3377/archive} tar -xvf hello.tar

hello

hello.c

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.tar hello.zip

**3. Use gzip to create .gz files with hello and hello.c, then restore the files**

{cslinux1:~/cs3377/archive} gzip hello hello.c

{cslinux1:~/cs3377/archive} ls

hello.c.gz hello.gz hello.tar hello.zip

{cslinux1:~/cs3377/archive} gzip -d hello.c.gz

{cslinux1:~/cs3377/archive} gzip -d hello.gz

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.tar hello.zip

**4. Compress hello.tar into hello.tar.gz using gzip**

{cslinux1:~/cs3377/archive} gzip hello.tar

{cslinux1:~/cs3377/archive} ls

hello hello.c hello.c.gz hello.gz hello.tar hello.tar.gz hello.zip

**5. Compare file sizes of different options**

{cslinux1:~/cs3377/archive} ls -l

total 408

-rw------- 1 dtp180003 sn 8512 Jul 4 18:54 hello

-rw------- 1 dtp180003 sn 100 Jul 4 18:54 hello.c

-rw------- 1 dtp180003 sn 122 Jul 4 18:54 hello.c.gz

-rw------- 1 dtp180003 sn 2379 Jul 4 18:54 hello.gz

-rw------- 1 dtp180003 sn 20480 Jul 4 19:08 hello.tar

-rw------- 1 dtp180003 sn 2626 Jul 4 18:58 hello.tar.gz

-rw------- 1 dtp180003 sn 2752 Jul 4 18:55 hello.zip

**6. Best compression observations**

Tar is the worst obviously, which is why it can be compressed using gzip. Tar.gz is slightly smaller than zip compression. Simply using gzip without first using tar results in separate files that are individually smallest, but it removes the convenience of having everything in one folder. For these reasons tar.gz is best, but zip is not much worse.

**Assignment 2 Part2**

**First, create a folder as shown below.**

\*\* Do this part using cs1.utdallas.edu

First, create a directory named cs3377 (if you have not created it yet). All of your work for cs3377 assignment(s) should be placed hcd ere for cs3377.

Second, create a subdirectory A2-netid in cs3377 directory (where netid is your UTD netid, for example, A2-rkm010300). All of your work for Assignment2 Part2 should be placed here.

You should be able to go to A2-netid directory by typing: cd $HOME/cs3377/A2-netid

**Use command(s) to create directories and files as shown below.**

Task#1.

After you created a directory named A2-netid in cs3377 (where all of your work for this part should be placed here), do the following commands to show that you have created in this directory.

cd $HOME/cs3377/A2-netid

date

whoami

uname -a

ls -lR

Copy and paste the run-log of your command and its output here (console output or screenshot for this task)

|  |
| --- |
| {cslinux1:~/cs3377} cd $HOME/cs3377/A2-dtp180003  {cslinux1:~/cs3377/A2-dtp180003} date  Mon Jul 6 12:23:57 CDT 2020  {cslinux1:~/cs3377/A2-dtp180003} whoami  dtp180003  {cslinux1:~/cs3377/A2-dtp180003} uname -a  Linux cslinux1.utdallas.edu 3.10.0-1127.8.2.el7.x86\_64 #1 SMP Tue May 12 16:57:42 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux  {cslinux1:~/cs3377/A2-dtp180003} ls -lR  .:  total 0 |

Task#2.

To create all the files and directories (all in A2-netid directory as shown below) from your c/c++ program as shown and explained below.

Design and implement C/C++ program (a2prog1.c and its executable named a2prog1) to create directories and files, and/or to navigate the directories as shown below. For this part, your program will use various calls (as provided in APUE Chapter 3 and Chapter 4 for various system API calls to do the following tasks: open, read, write, chdir, mkdir, stat, chmod, link, access, symlink, getcwd, etc.). For example, in your program, use "getcwd" call to find out your current working directory, "chdir" call to change your current working directory, "mkdir" call to create a directory, "open" to create a new file, etc. For time-date format and output, see APUE 6.10. For softlink (create and read), see APUE 4.18. Note. All the sample codes in the figures of APUE are available in APUE code (apue.3e => figlinks).

For your output, your program should print a brief message with a time-of-day timestamp (e.g., " \*\* To create dir1 at 2019.09.29 13:59 \*\*") before each call (e.g., to create a file or to create a directory). See APUE Ch6.10 (Figure 6.11 prog to get and display the time of day).

Warning. Your program should not use any of "system( … )" calls for this part (otherwise, your grade for this part will be 0).

file1.txt

file2.txt

link5

file4.txt

file3.txt

# Part 2 Solution

Task#2

1a. List of your program: a2prog1.c

|  |
| --- |
| #include "apue.h"  #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include <fcntl.h>  int  main(void)  {  time\_t t;  struct tm \*tmp;  char buf1[16];  char buf2[64];  time(&t);  tmp = localtime(&t);  strftime(buf2, 64, "create dir1 at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  size\_t size;  char \*ptr = path\_alloc(&size);  if (getcwd(ptr, size) == NULL)  err\_sys("getcwd failed");  if((mkdir("dir1",00777))==-1) {  fprintf(stdout,"error in creating dir\n");  }  if(chdir("dir1") < 0)  err\_sys("chdir failed");  if(open("file1.txt",O\_CREAT | O\_WRONLY) < 0)  err\_sys("open file1.txt failed");  else{  strftime(buf2, 64, "create file1.txt at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  }  if(open("file2.txt",O\_CREAT | O\_WRONLY) < 0)  err\_sys("open file2.txt failed");  else{  strftime(buf2, 64, "create file2.txt at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  }  if(chdir(ptr) < 0)  err\_sys("chdir failed");  time(&t);  tmp = localtime(&t);  strftime(buf2, 64, "create dir2 at %r, %a %b %d, %Y", tmp);  printf("%s\n",buf2);  if((mkdir("dir2",00777))==-1) {  fprintf(stdout,"error in creating dir\n");  }  if(chdir("dir2") < 0)  err\_sys("chdir failed");  if(open("file3.txt",O\_CREAT | O\_WRONLY) < 0)  err\_sys("open file3.txt failed");  else{  strftime(buf2, 64, "create file3.txt at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  }  if(open("file4.txt",O\_CREAT | O\_WRONLY) < 0)  err\_sys("open file4.txt failed");  else{  strftime(buf2, 64, "create file4.txt at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  }  if(chdir(ptr) < 0)  err\_sys("chdir failed");  time(&t);  tmp = localtime(&t);  strftime(buf2, 64, "create dir3 at %r, %a %b %d, %Y", tmp);  printf("%s\n",buf2);  if((mkdir("dir3",00777))==-1) {  fprintf(stdout,"error in creating dir\n");  }  if(symlink("dir2/file4.txt","dir3/link5") < 0)  err\_sys("create symlink failed");  else{  strftime(buf2, 64, "create link5 at %r, %a %b %d, %Y", tmp);  printf("%s\n", buf2);  }  exit(0);  } |

1b. List of your program run (session/program-run log/history)

|  |
| --- |
| {cslinux1:~/cs3377/A2-dtp180003} ./a2prog1  create dir1 at 01:58:39 PM, Mon Jul 06, 2020  create file1.txt at 01:58:39 PM, Mon Jul 06, 2020  create file2.txt at 01:58:39 PM, Mon Jul 06, 2020  create dir2 at 01:58:39 PM, Mon Jul 06, 2020  create file3.txt at 01:58:39 PM, Mon Jul 06, 2020  create file4.txt at 01:58:39 PM, Mon Jul 06, 2020  create dir3 at 01:58:39 PM, Mon Jul 06, 2020  create link5 at 01:58:39 PM, Mon Jul 06, 2020 |

1c. List of your validation with "ls -lR" command to list all its files and directories.

|  |
| --- |
| {cslinux1:~/cs3377/A2-dtp180003} ls -lR  .:  total 216  -rwx--x--x 1 dtp180003 sn 14072 Jul 6 13:58 a2prog1  -rw------- 1 dtp180003 sn 2060 Jul 6 13:58 a2prog1.c  drwx--x--x 2 dtp180003 sn 54 Jul 6 13:58 dir1  drwx--x--x 2 dtp180003 sn 54 Jul 6 13:58 dir2  drwx--x--x 2 dtp180003 sn 23 Jul 6 13:58 dir3  ./dir1:  total 48  -r-s------ 1 dtp180003 sn 0 Jul 6 13:58 file1.txt  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file2.txt  ./dir2:  total 48  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file3.txt  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file4.txt  ./dir3:  total 24  lrwxrwxrwx 1 dtp180003 sn 14 Jul 6 13:58 link5 -> dir2/file4.txt |

2a. List of your program: a2prog2.c

|  |
| --- |
|  |

2b. List of your program run (session/program-run log/history)

|  |
| --- |
|  |

3. List of Makefile

|  |
| --- |
|  |

Task#3.

After Task#2 is done, provide the result of the following commands here to show that you have done this part.

cd $HOME/cs3377/A2-netid

date

whoami

uname -a

ls -lR

Copy and paste the run-log of your command and its output here (console output for this task)

|  |
| --- |
| {cslinux1:~} cd $HOME/cs3377/A2-dtp180003  {cslinux1:~/cs3377/A2-dtp180003} date  Mon Jul 6 14:01:46 CDT 2020  {cslinux1:~/cs3377/A2-dtp180003} whoami  dtp180003  {cslinux1:~/cs3377/A2-dtp180003} uname -a  Linux cslinux1.utdallas.edu 3.10.0-1127.8.2.el7.x86\_64 #1 SMP Tue May 12 16:57:42 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux  {cslinux1:~/cs3377/A2-dtp180003} ls -lR  .:  total 216  -rwx--x--x 1 dtp180003 sn 14072 Jul 6 13:58 a2prog1  -rw------- 1 dtp180003 sn 2060 Jul 6 13:58 a2prog1.c  drwx--x--x 2 dtp180003 sn 54 Jul 6 13:58 dir1  drwx--x--x 2 dtp180003 sn 54 Jul 6 13:58 dir2  drwx--x--x 2 dtp180003 sn 23 Jul 6 13:58 dir3  ./dir1:  total 48  -r-s------ 1 dtp180003 sn 0 Jul 6 13:58 file1.txt  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file2.txt  ./dir2:  total 48  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file3.txt  ---S------ 1 dtp180003 sn 0 Jul 6 13:58 file4.txt  ./dir3:  total 24  lrwxrwxrwx 1 dtp180003 sn 14 Jul 6 13:58 link5 -> dir2/file4.txt |

Submit two files: (1) this word document with your solution and (2) a zip file of A2-netid (containing all the programs, output files, and log of program run).

Warning. Your program should not use any of "system( … )" calls for this part (otherwise, your grade for this part will be 0).

Any "poor" documentation (this document with your answer) may result in a penalty (up to 10%).

Any "missing" submission of the files (including the zip file for this part) may result in 0 for this assignment grade.

\*\* A demo will be announced and scheduled by TA. For any scheduling conflict, please contact or consult with TA (or the instructor) for your situation, and/or for alternate time or discretion as soon as possible, before the due.

Task#3

After Task#2 is done, do the following commands to show that you have done all of Task#2.

cd $HOME/cs3377/A2-netid

date

whoami

uname -a

ls -lR

Copy and paste the run-log of your command and its output here (console output or screenshot for this task)

|  |
| --- |
|  |