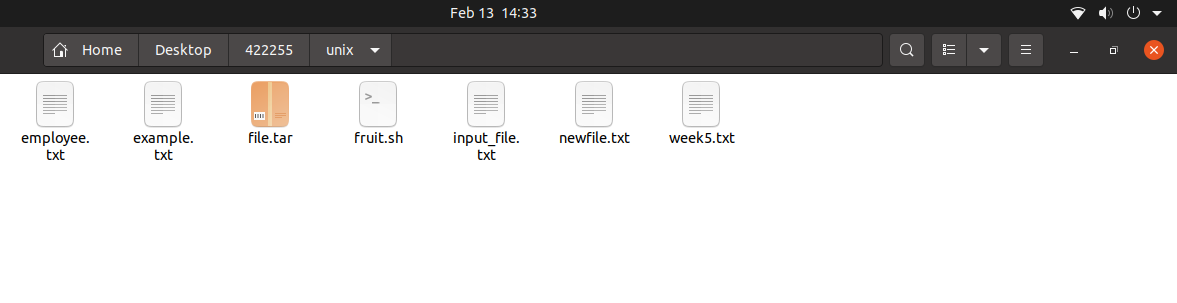
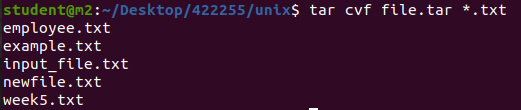
### **ASSIGNMENT-06: (Week-06):**

Implement shell scripts containing tar and cpio commands.

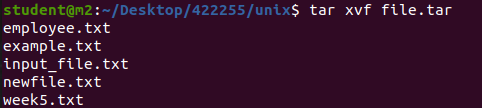
1. **Tar command:**

### **1. Creating an uncompressed tar Archive using option -cvf**



### **2. Extracting files from Archive using option -xvf**

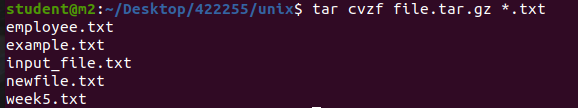
* ‘-x’: Extracts files from an archive.
* ‘-v’: Displays verbose output during the extraction process.
* ‘-f’: Specifies the filename of the archive.



**3. gzip compression on the tar Archive, using option -z**

This command creates a tar file called file.tar.gz which is the Archive of .txt files.

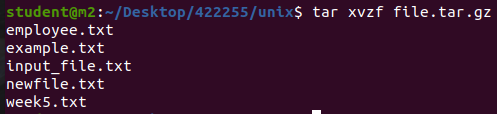
tar cvzf file.tar.gz \*.txt



* ‘-z’: Uses gzip compression.
* ‘-j’: Uses bzip2 compression.
* ‘-J’: Uses xz compression.

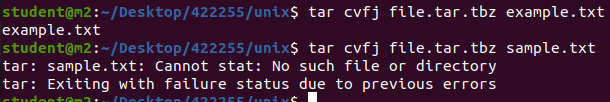
### **4. Extracting a gzip tar Archive \*.tar.gz using option -xvzf :**

This command extracts files from tar archived file.tar.gz files. tar xvzf file.tar.gz



### **5. Creating compressed tar archive file in Linux using option -j**

This command compresses and creates archive files less than the size of the gzip. Both compress and decompress take more time than gzip. tar cvfj file.tar.tbz example.cpp

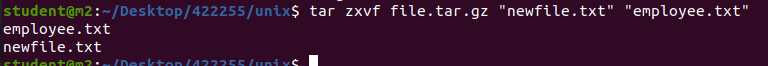


### **6. Untar multiple .tar, .tar.gz, .tar.tbz file in Linux:**

This command will extract or untar multiple files from the tar, tar.gz and tar.bz2 archive file. For example, the above command will extract “fileA” “fileB” from the archive files.

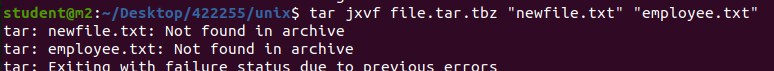
tar xvf file.tar "fileA" "fileB"

or

tar zxvf file1.tar.gz "fileA" "fileB"

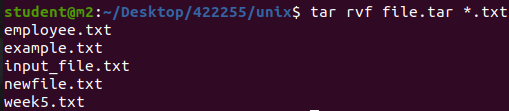
or

tar jxvf file2.tar.tbz "fileA" "fileB"



### **7. Update existing tar file in Linux**

tar rvf file.tar \*.c



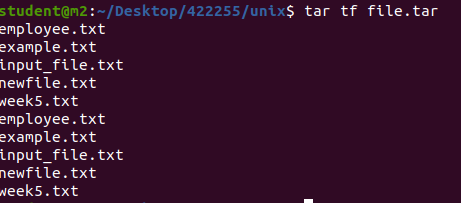
**Output:**

os1.c

### **8. List the contents and specify the tarfile using option -tf**

This command will list the entire list of archived files. We can also list for specific content in a tarfile

tar tf file.tar



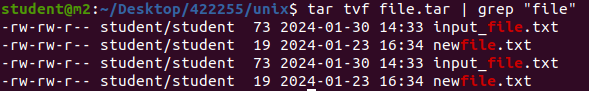
**Output:**

example.cpp

### **9. Applying pipe to through ‘grep command’ to find what we are looking for**

This command will list only for the mentioned text or image in grep from archived file.

tar tvf file.tar | grep "*text to find*"

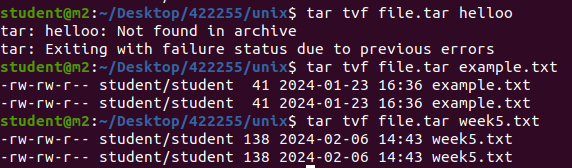


or

tar tvf file.tar | grep "*filename.file extension*"

### **10. We can pass a file name as an argument to search a tarfile:**

This command views the archived files along with their details.

tar tvf file.tar filename

### **11. Viewing the Archive using option -tvf**

### tar tvf file.tar

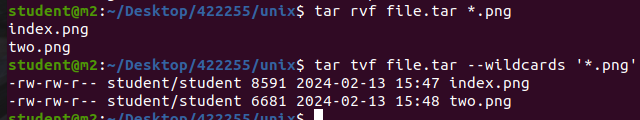
**Output:**

-rwxrwxrwx root/root 191 2017-09-17 02:20 os2.c

-rwxrwxrwx root/root 218 2017-09-17 02:20 os3.c

-rwxrwxrwx root/root 493 2017-09-17 02:20 os4.c

### **12. To search for an image in .png format:**

This will extract only files with the extension .png from the archive file.tar. The -wildcards option tells tar to interpret wildcards in the name of the files to be extracted; the filename (\*.png) is enclosed in single-quotes to protect the wildcard (\*) from being expanded incorrectly by the shell. $ tar tvf file.tar --wildcards '\*.png'

1. **Cpio command:**

**cpio** stands for “**copy in, copy out**“. It is used for processing the archive files like *\*.cpio* or *\*.tar*. This command can copy files to and from archives. **Synopsis:**

* **Copy-out Mode:** Copy files named in name-list to the archive **Syntax:** cpio -o < name-list > archive
* **Copy-in Mode:** Extract files from the archive **Syntax:** cpio -i < archive
* **Copy-pass Mode:** Copy files named in name-list to destination-directory **Syntax:** cpio -p destination-directory < name-list

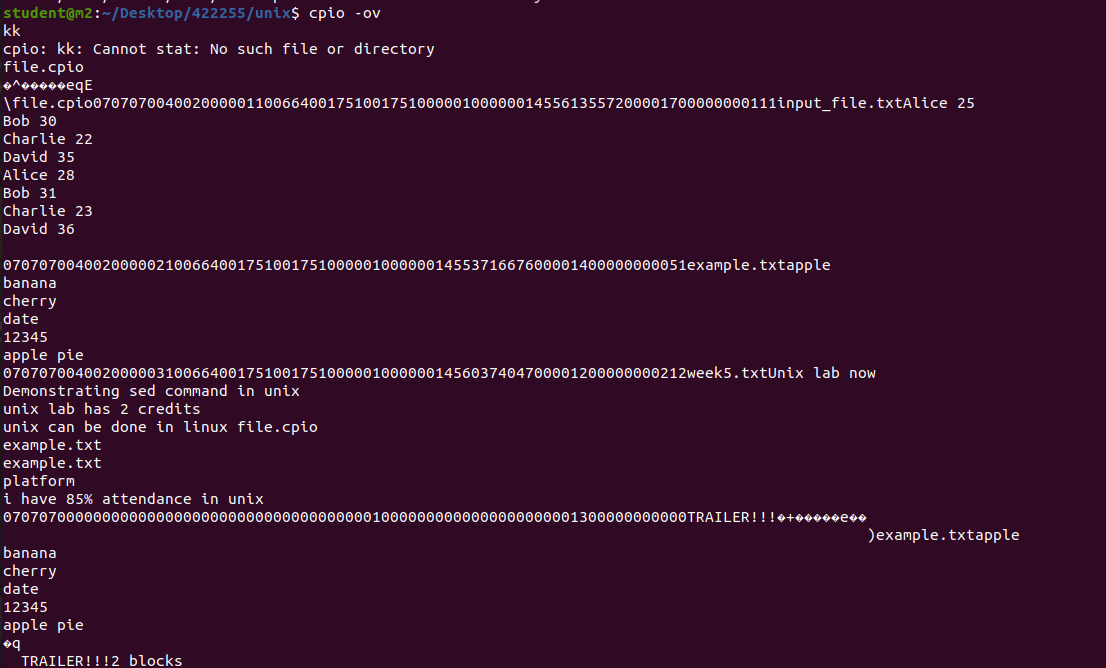
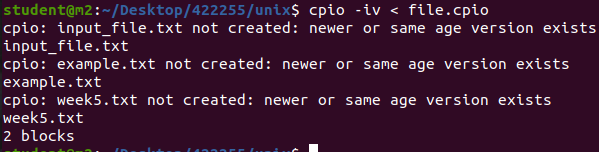
**Policy Options:**

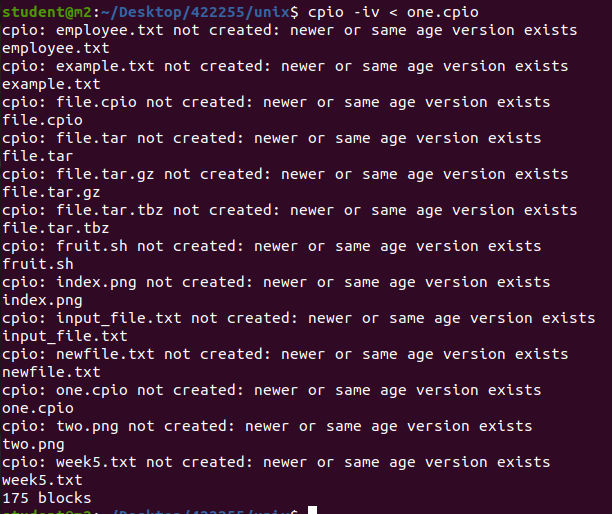
* **-i, –extract:** Extract files from an archive and it runs only in copy-in mode.
* **-o, –create:** Create the archive and it runs only in copy-out mode.
* **-p, –pass-through:** Run in copy-pass mode.
* **-t, –list:** Print a table of contents of all the inputs present.

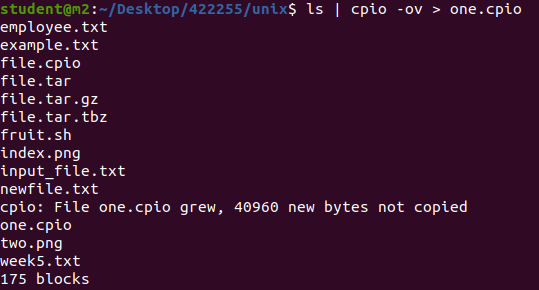
**Operation modifiers valid in any Mode:**

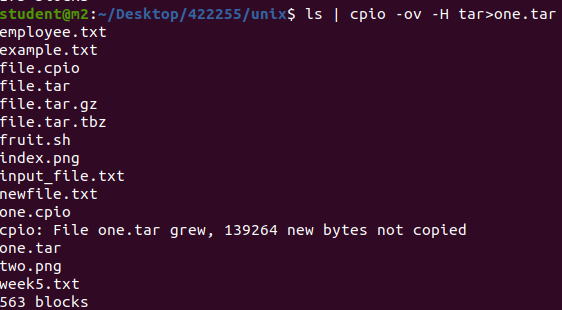
* **-B:** Changes the I/O block size to 5120 bytes.
* **-c:** Use the old portable (ASCII) archive format.
* **-C, –io-size=NUMBER:** Set the I/O block size to the given particular NUMBER of bytes.
* **-D, –directory=DIR:** Changes to Directory *DIR*.
* **-H, –format=FORMAT:** Use given arc.
* **-v, –verbose:** List the files processed in a particular task.
* **-V, –dot:** Print “.” for each file processed in a particular task.
* **-W, –warning=FLAG:** Control warning display. Currently FLAG is one of ‘*none*‘, ‘*truncate*‘, ‘*all*‘.

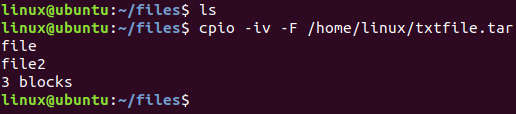
**Examples:**

* **To create a \*.cpio file :** We can create \*.cpio files containing files and directory with the help of cpio command. **Syntax:** cpio -ov < name-list > archive . Here *-ov* is used as -o create the new archive and -v list the files processed. 
* **To extract a \*.cpio file:** We can extract *\*.cpio* files containing files and directory with the help of cpio command. **Syntax:** cpio -iv < archive  
   



* **To create \*.tar archive file using cpio:** The cpio helps to create a *\*.tar* archive. **Syntax:** cpio -ov -H tar > archive  
   
* ls | cpio -ov -H tar>one.tar



* **To extract \*.tar archive file using cpio:** The cpio helps to extract \*.tar files containing files and directory. **Syntax:** cpio -iv -F < archive  
   
* **To create a \*.cpio archive with selected files:** We can create *\*.cpio* files containing specific files with the help of cpio command. In the example we are using *.txt* files. **Syntax:** find . -iname "\*.txt" | cpio -ov > archive  
   