

```
echo MENU
echo 1: Add
echo 2: Subtract
echo 3: Multiply
echo 4: Divide
echo 5: Exit
echo Enter Your Choice
read ch
echo Enter First No.
read x
echo Enter Second No.
read y
case $ch in
1) z=`expr $x + $y`
   echo Sum is $z;;
2) z=`expr $x - $y`
   echo Subtraction is $z;;
3) z=`expr $x \* $y`
   echo Multiplication is $z;;
4) z=`expr $x / $y`
   echo Divition is $z;;
5) exit;;
*) Invalid Choice
esac
```

I Pro1.sh 25/25 100%

← → ⌂ bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
echo Enter the name of file
read fn
path=$(find "/$HOME" -name "$fn")
echo "The Complete path of the file '$fn' is '$path'"
```

卷之三

- Pro2.sh 1/5 20%



```
echo MENU
echo 1:Sort a file and store the sorted file in a new file
echo 2:Execute more than one command without affecting results
echo 3:Display version of Linux
echo 4:Get Online help of a command
echo 5:Exit
echo Enter your choice
read ch
case $ch in
1) echo "Enter the name of the file to be sorted"
   read fn
   echo "Enter the name of the file to store the sorted file of $fn"
   read fn2
   sort "$fn" > "$fn2"
   cat "$fn2";;
2) date; ls; echo "HELLO";;
3) echo "Version of Linux"
   uname -r;;
4) echo Enter the name of command for online help
   read cmnd
   man "$cmnd";;
5) exit;;
*) echo "Invalid Choice"
   exit 1;;
esac
```

```
echo MENU
echo 1:Display hidden files
echo 2>Delete directory
echo 3:Interactive copying
echo 4:Interactive delete
echo 5:Use of mv command
echo 6:Exit
echo Enter your choice
read ch

case $ch in
1) ls -all;;
2) echo Enter the name of the directory to be deleted
   read dirname
   rm -r $dirname;;
3) echo Enter the name of the file to be copied
   read src
   echo Enter the name of the destination file
   read dest
   cp -i $src $dest;;
4) echo Enter the name of the file to be deleted
   read fn
   if [ -e $fn ]
   then
      rm -i $fn
   else
      echo file $fn does not exist
   fi;;
5) echo Enter the source file name
   read src
   echo Enter the destination file name
   read dest
   mv $src $dest;;
6) exit;;
*) echo Invalid Choice
   exit;;
- Pro4.sh 1/38 2%
```

```
echo MENU
echo 1:Create a text file with name and age
echo 2:Display the contents of the file on screen
echo 3:Delete the directories mydir and newdir in one shot
echo 4:Sort a numeric file
echo 5:Change the permission of the file network to 666
echo 6:Exit

echo Enter your choice
read ch

case $ch in
1) echo Enter the name of the new file
   read fn
   cat > $fn
   cat $fn;;
2) echo Enter the name of file for Displaying the contents of the file
   read fn
   cat $fn;;
3) rmkdir mydir newdir;;
4) echo Enter the name of the numeric file
   read fn1
   sort -n $fn1;;
5) chmod 666 network;;
6) exit;;
*) echo Invalid choice;;
esac
```

~
~
~
~
~
~
~
~
~

- Pro5.sh 1/28 3%

```
echo Enter the file name
read fn
if [ -e $fn ]
then
    date -r $fn
else
    echo $fn file does not exist
fi
```

```
echo Enter first latter of username  
reach un  
who | cut -d " " -f1 | grep "$un"  
cnt=$(who | cut -d " " -f1 | grep -c "$un")  
echo "$cnt users with userame starting with `\$un`"
```

```
for file in `ls`
do
if [ -s $file ]
then
echo file $file is not empty, skipping
else
echo Removing zero sized file:$file
rm -i $file
fi
done
```

JSLinux

bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
#Loop through all files in the current directory for that first of all change any file in executable mode use chmod 777 filename
for file in `ls`
do
if [ -x $file ]
then
echo "$file"
fi
done
```

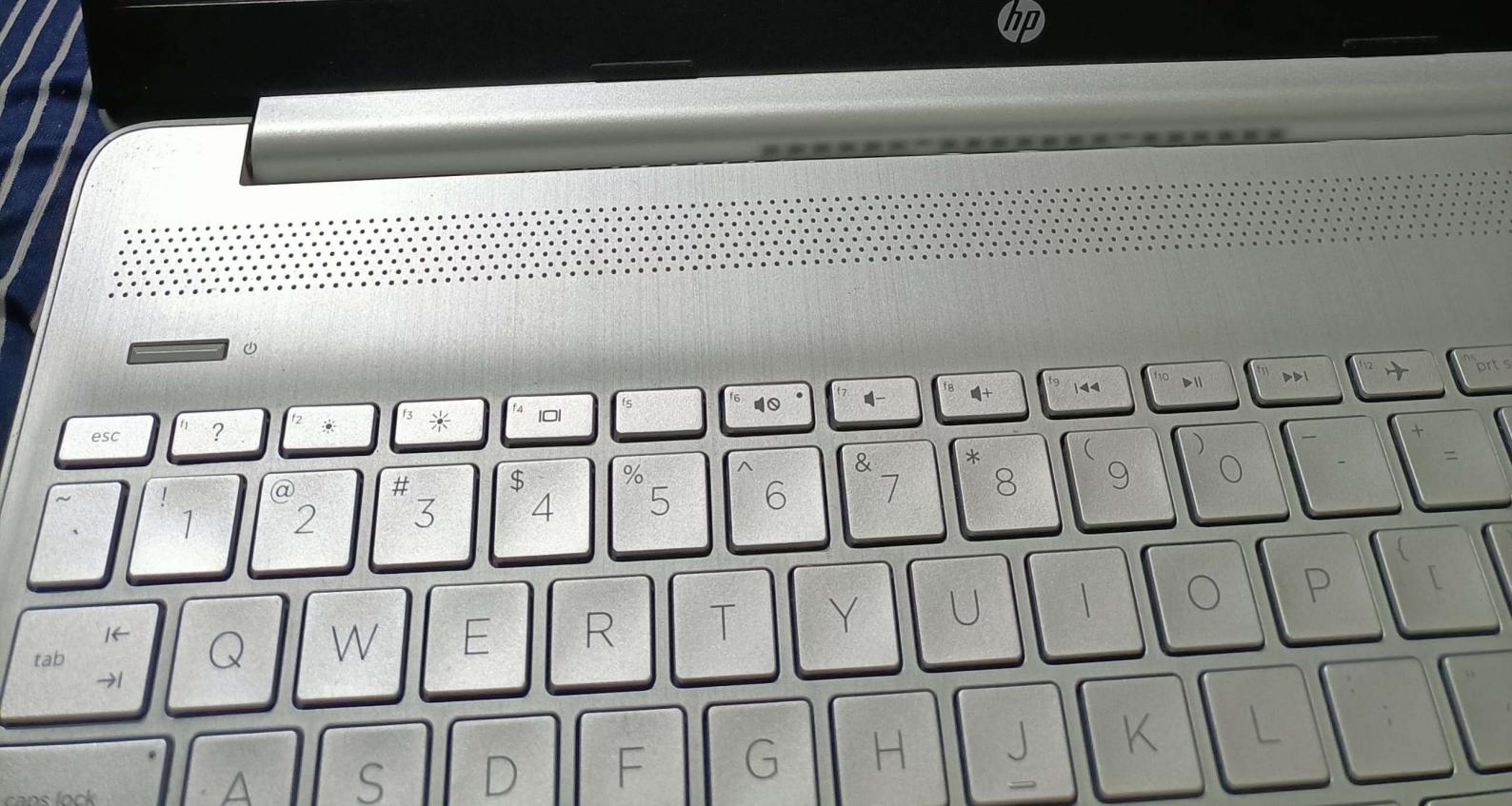
I Pro9.sh [Modified] 1/9 11%

26°C Smoke

© 2011-2021 Fabrice Bellard - [News](#) - [VM list](#) - [FAQ](#) - [Technical notes](#)

Search

hp



```
time=$(date "+%H")
user=$LOGNAME

if [ $time -ge 0 -a $time -lt 11 ]
then
    echo Good Morning, $user
elif [ $time -ge 11 -a $time -lt 16 ]
then
    echo Good Afternoon, $user
elif [ $time -ge 16 -a $time -le 20 ]
then
    echo Good Evening, $user
else
    echo Hello! $user , welecome to this session
fi
```

```
total=`ls -all | wc -l`
echo Total files and directory in current directory: $total
f=0
d=0
for i in `ls`
do
  if [ -d $i ]
  then
    d=`expr $d + 1`
  else
    f=`expr $f + 1`
  fi
done
echo Number of files $f
echo Number of Directories $d
```

```
filename=$1
if [ -e $filename ]
then
  if [ -f $filename ]
  then
    echo $filename is a regular file
    cat $filename
  else
    echo $filename exists but is not a regular file, cannot display.
  fi
else
  echo File $filename does not exist.
fi
```

```
echo Enter the filename  
read fn  
if [ -d $fn ]  
then  
    echo $fn is a directory file  
    echo contents of directory $fn  
    ls $fn  
else  
    echo File is not a directory file or does not exist  
fi
```

```
n=$1
echo $fn

if [ -e $fn ]
then
    echo File is exists.
    if [ -x $fn ]
    then
        echo File is executable
    else
        echo File $fn exists but is not executable.
    fi
else
    echo File $fn does not exist.
fi
```

→ C bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
echo current directory is `pwd`  
for dir in `ls`  
do  
    if [ -d $dir ]  
    then  
        echo content of subdirectory $dir  
        ls $dir  
    fi  
done
```

```
total=`ls -all | wc -l`  
echo Total files and directories are: $total  
f=0  
d=0  
for i in `ls`  
do  
    if [ -d $i ]  
    then  
        d=`expr $d + 1`  
    else  
        f=`expr $f + 1`  
    fi  
done  
echo Total files are $f  
echo Total Directories are $d
```

→ G bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
echo Enter first file name
read f1
echo Enter second file name
read f2
if [ -f $f1 ]
then
    if [ -f $f2 ]
    then
        echo Both files exists
        echo Appending contents of $f2 into $f1
        cat $f2 >> $f1
        cat $f1
    else
        echo file $f2 does not exist
    fi
else
    echo file $f1 does not exist
fi
```

- Pro17.sh 1/19 5%



← → C bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
echo Displaying the contents of file:$1  
more $1
```

```
echo Copy the contents of the first file to the second file  
cp $1 $2
```

```
echo File copied sucessfully.
```

```
echo Displaying the contents of copied file:$2  
cat $2
```



```
while true
do
    echo MENU
    echo 1:Calender of the current month and year
    echo "2:Display greeting (Good Morning/Good Afternoon/Good Evening)"
    echo 3:User name and home directory
    echo 4:Terminal name and type
    echo 5:Machine name
    echo 6:Number of users currently logged in and list of users
    echo 7:Exit
    echo Enter your choice:
    read ch

    case $ch in
        1) cal;;
        2) hour=$(date +%H)
            if [ $hour -ge 0 -a $hour -lt 12 ]
            then
                echo "Good Morning"
            elif [ $hour -ge 12 -a $hour -lt 18 ]
            then
                echo "Good Afternoon"
            else
                echo "Good Evening"
            fi;;
        3) echo User name: $(whoami)
            echo Home directory: $HOME;;
        4) echo Terminal name: $TERM
            echo Terminal type: $TERM_PROGRAM;;
    I Pro19.sh [Modified] 1/50 2%
```



1 86°F
Smoke



Search



← → C bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
case $ch in
1) cal;;
2) hour=$(date +%H)
   if [ $hour -ge 0 -a $hour -lt 12 ]
   then
      echo "Good Morning"
   elif [ $hour -ge 12 -a $hour -lt 18 ]
   then
      echo "Good Afternoon"
   else
      echo "Good Evening"
   fi;;
3) echo User name: $(whoami)
   echo Home directory: $HOME;;
4) echo Terminal name: $TERM
   echo Terminal type: $TERM_PROGRAM;;
5) echo Machine name: $HOSTNAME;;
6) echo Number of user currently logged in: `who | wc -l`  
echo "List of users currently logged in:" who ;;
7) echo Exiting..
   break;;
*) echo Invalid choice. Please enter a number from 1 to 7.;;
esac
done
```

I Pro19.sh [Modified] 50/50 100%



```
while true
do
    echo MENU
    echo 1:Concatenate two strings
    echo 2:Rename a file
    echo 3:Delete a file
    echo 4:Copy a file to a specific location
    echo 5:Exit
    echo Enter your choice
    read ch

    case $ch in
        1) echo Enter the first string
            read str1
            echo Enter the second string
            read str2
            echo Concatenated strings: $str1$str2 ;;

        2) echo Enter the current file name
            read old_name
            echo Enter the new file name
            read new_name
            mv $old_name $new_name
            echo File renamed successfully. ;;

        3) echo Enter the file name to delete:
            read file_to_delete
            rm $file_to_delete
            echo File deleted successfully. ;;

        4) echo Enter the file name to copy:
            read file_to_copy
            echo Enter the destination directory:
            read destination
- Pro20.sh 1/48 2%
```



```
1) echo Enter the first string
   read str1
   echo Enter the second string
   read str2
   echo Concatenated strings: $str1$str2 ;;

2) echo Enter the current file name
   read old_name
   echo Enter the new file name
   read new_name
   mv $old_name $new_name
   echo File renamed successfully. ;;

3) echo Enter the file name to delete:
   read file_to_delete
   rm $file_to_delete
   echo File deleted successfully. ;;

4) echo Enter the file name to copy:
   read file_to_copy
   echo Enter the destination directory:
   read destination
   cp $file_to_copy $destination
   echo File copied successfully. ;;

5) echo Exiting
   break ;;

*) echo Invalid choice. Please enter a number from 1 to 5. ;;

esac
```

done

```
for file in *.txt
do
    new_name="${file%.txt}.dat"
    mv $file $new_name
done
echo Suffix changed successfully.
```

I Pro21.sh [Modified] 1/8 12%

```
if [ $# -eq 0 ]
then
    directory=$HOME
else
    directory=$1
fi

if [ -d $directory ]
then
    echo "Contents of $directory:"
    ls $directory
else
    echo Directory $directory does not exist.
fi
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

```
for file in `"$HOME`/c*  
do  
    if [ -f $file ]  
    then  
        new_name="${file}111"  
        mv $file $new_name  
        echo "Renamed $file to $new_name"  
    fi  
done
```

```
#If you use -s option, cmp doesn't show this information about where the file differ.  
#Instead, it just gives an exit status  
  
file1=$1  
file2=$2  
  
if cmp -s $file1 $file2  
then  
    echo Contents of $file1 and $file2 are the same  
    echo Deleting $file2  
    rm $file2  
else  
    echo Contents of $file1 and $file2 are different  
fi
```

I Pro24.sh [Modified] 1/15 6%



90°F
Smoke



Search



```
source_dir=$1
dest_dir=$2

if [ ! -d $source_dir ]
then
    echo "Error: Source directory $source_dir does not exist."
    exit 1
fi

if [ ! -d $dest_dir ]
then
    mkdir $dest_dir
    echo "Destination directory $dest_dir created."
fi

echo Copy files from source $source_dir to destination $dest_dir
cp -r $source_dir/* $dest_dir
echo Files copied from $source_dir to $dest_dir successfully
ls $dest_dir
```

- Pro25.sh 1/21 4%



90°F



Search

```
while true
do
    echo MENU
    echo 1: List home directory
    echo 2: Date
    echo 3: Print working directory
    echo 4: Users logged in
    echo 5: Exit
    echo Enter your choice
    read ch

    case $ch in
        1) ls "$HOME" ;;
        2) date ;;
        3) pwd ;;
        4) who ;;
        5) echo Exiting
            break ;;
        *) echo Invalid choice ;;
    esac
```

done

~
~
~
~

- Pro26.sh 1/31 3%



90°F
Smoke

```
echo Enter the directory name:  
read dir  
  
if [ -d $dir ]  
then  
    echo Hidden files in $dir:  
    ls -a $dir | grep "\."  
else  
    echo Directory $dir does not exist  
fi
```

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

I Pro27.sh [Modified] 1/12 8%



```
file1=$1
file2=$2
output_file=$3

paste $file1 $file2 > $output_file
echo Combine horizontally:
cat $output_file

echo "_____"

cat $file1 $file2 >> $output_file
echo combine Vertically:█
cat $output_file
```

I Pro28.sh [Modified] 13/16 81%

```
filename=$1

#tr is the command used for translating or deleting characters.
#-d option tells -tr to delete characters ' '.
#< $filename It read ths contents of file.

tr -d ' ' < $filename > $filename.tmp

echo File content after deleting spaces:
cat $filename.tmp
```

I Pro29.sh [Modified] 6/11 54%



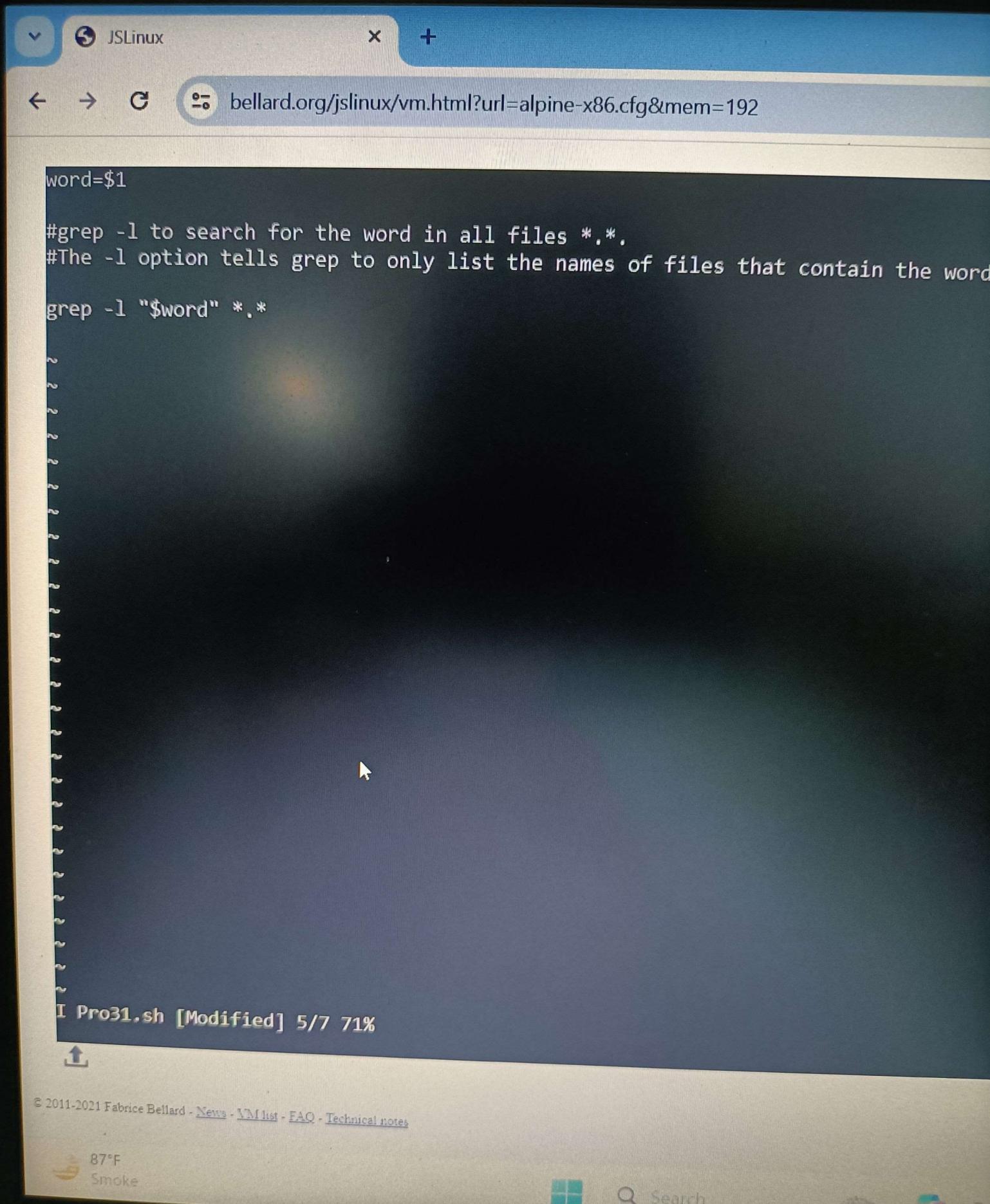
```
echo Enter the date in yyyy-mm-dd format
read input_date

#Get the day name (e.g Monday,Tuesday) for the input data
day=$(date -d "$input_date" +%A)

#check if the day falls on a weekday (1 to 5) or a weekend (6 or 7)
if [[ $day == "Saturday" || $day == "Sunday" ]]
then
    echo $input_data falls on a weekend $day.
else
    echo $input_data falls on a weekday $day.
fi
```

I Pro30.sh [Modified] 1/15 6%





```
echo The last Modification file  
ls -lt | head -2 | tail -1
```

- Pro32.sh 1/3 33%



chat.openai.com/



ChatGPT 3.5



ChatGPT

The `stat` command is used to display detailed information about a file, including its permissions. The `-c` option is used to specify the output format for `stat`. In this case, `%A` is a format specifier that represents the file's permissions in the form of a string (e.g., `-rwxr-xr--`).

Here's a breakdown of the format specifier `%A`:

- `'%A'`: Display the access permissions of the file in human-readable form.

When you run `stat -c "%A"`

Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.



```
filename=$1

if [ -e $filename ]
then
    echo "Permissions of $filename:"
    ls -all $filename | cut -c 1-10
else
    echo file does not exists
fi
```

```
echo MENU
echo 1: Display the calander of month m1 and m2
echo 2:Display the calender of the months from m1 to m2
echo Enter your choice
read ch

case $ch in
1) cal $1 `date "+%y"`;
   cal $2 `date "+%y"`;
;;
2) i=$1
   while [ $i -le $2 ]
   do
      clear
      cal $i `date "+%y"`;
      sleep 2
      i=`expr $i + 1`
   done;;
*) echo Invalid choice ;;
esac
```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

- Pro34.sh 1/21 4%



```
filename=$1

echo MENU
echo 1: Diaplay all the words of the file in ascending order
echo 2: Display the file in descending order
echo 3: Tpggle all the characters in the file
echo 4: Display the type of the file
echo Enter your choice
read ch

case $ch in

1) echo "Words in ascending order:"
   #'tr' replace '-s' muliple characters '[:space:]' whitespace with a single '\n' new line.
   cat $filename | tr -s '[:space:]' '\n' | sort ;;

2) echo File in descending order
   #'tac' which is 'cat' spelled backward is used to display the contents of file in descending order
   tac $filename ;;

3) echo Toggling all characters in the file
   tr '[:upper:][:lower:]' '[:lower:][:upper:]' < $filename ;;

4) echo Type of the file:
   file $filename ;;

*) echo Invalid choice;;
esac
```

~
~
~
~
~
~
~
I Pro35.sh [Modified] 19/30 63%



```
username=$1

if who | grep "^$username"
then
    echo $username is logged in
else
    echo $username is not logged in
fi
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

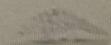
~

~

~

~

I Pro36.sh [Modified] 3/8 37%



```
filename=$1

echo MENU
echo 1: Diaplay all the words of the file in ascending order
echo 2: Display the file in descending order
echo 3: Tpggle all the characters in the file
echo 4: Display the type of the file
echo Enter your choice
read ch

case $ch in
1) echo "Words in ascending order:"
   #'tr' replace '-s' muliple characters '[:space:]' whitespace with a single '\n' new line.
   cat $filename | tr -s '[:space:]' '\n' | sort ;;
2) echo File in descending order
   #'tac' which is 'cat' spelled backward is used to display the contents of file in descending order
   tac $filename ;;
3) echo Toggling all characters in the file
   tr '[:upper:][:lower:]' '[:lower:][:upper:]' < $filename ;;
4) echo Type of the file:
   file $filename ;;
*) echo Invalid choice;;
esac
```

~
~
~
~
~
~
~

- Pro37.sh 1/30 3%



JSLinux

bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

```
#!/bin/sh
#'-c' count the number of lines in the '/etc/passwd' file that match the pattern '^.*:'
total_users=$(grep -c '^.*:' /etc/passwd)

logged_in_user=$(who | wc -l)

echo Total number of user: $total_users
echo Number of users currently logged in: $logged_in_user
```

I Pro38.sh [Modified] 1/8 12%

© 2011-2021 Fabrice Bellard - News - VM list - FAQ - Technical notes

79°F
Smoke



Search



```
filename=$1
size=$(ls -l $filename | awk '{print $5}')
date=$(ls -l $filename | awk '{print $6, $7, $8}')
protection=$(ls -l $filename | awk '{print $1}')
owner=$(ls -l $filename | awk '{print $3}')
echo $size $date $protection $owner
```

~
~
~
~
~
~
~
~
~
~
~
~

```
du -a | sort -n -r | head -n 5
```

bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192

filename=\$1

```
tr '[:upper:][:lower:]' '[:lower:][:upper:]' < $filename  
cat $filename
```

卷之三

- Pro41.sh 2/5 40%



15

77°F
Smoke



```
#file have data like 101 Junu 88
student_file=$1

echo Contents of the file sorted by marks in descending order:
sort -k3 -r $student_file

echo Name of students in alphabetical order
sort -k2 $student_file | awk '{print $2}'

echo Student accoring to their roll number
sort -k1 $student_file

echo Sorting file according to the second field and saving it to another file
sort -k2 $student_file > file1
cat file1

echo List of students who scored between 70 and 80
awk '$3 >= 70 && $3 <= 80' $student_file
```

I PPro45.sh [Modified] 2/19 10%



1 76°F
Smoke



Search

