

Shell Scripting 2020: Week 4

Stefan Ciprian Voinea
Student number: 015383372

November 29, 2020

31. ASCII art

Output of the execution:

```
cip ~/Desktop/UNI/ShellScripting2020/Week4 master ./task31_shortcat.sh shortcatz
That's not a number chief
x cip ~/Desktop/UNI/ShellScripting2020/Week4 master ./task31_shortcat.sh 5
# #
@
_|_
cip ~/Desktop/UNI/ShellScripting2020/Week4 master ./task31_shortcat.sh 10
# #
@
_|_
cip ~/Desktop/UNI/ShellScripting2020/Week4 master ./task31_shortcat.sh 500000
# #
@
```

Contents of the task31_shortcat.sh file:

```
#!/bin/bash

belly_lines=$1
shortcat="task31_shortcat.txt"

re='^[0-9]+$'
if ! [[ $belly_lines =~ $re ]]
then
    echo "That's not a number chief"
    exit 1
fi

if [ $belly_lines -gt 1 ]
then
    head -n 8 $shortcat
    for i in $(seq 1 $belly_lines)
    do
        sed "9q;d" $shortcat
    done
    tail -n 6 $shortcat
else
    echo "Nope"
fi
```

32. Plotting

Contents of the task32_create_random_data.sh file:

```
#!/bin/bash

output_file="task32_create_random_data.txt"

rm -f $output_file

for i in $(seq 1 $1)
do
    # echo $ $RANDOM % 10 $ $RANDOM % 10 >> $output_file
    echo $RANDOM $RANDOM >> $output_file
done

cat $output_file

# Plotting the data
gnuplot task32_create_random_data.p
```

Contents of the task32_create_random_data.p file:

```
set terminal png size 1000,1000
set output 'task32_create_random_data.png'
set title 'Random Numbers'
plot 'task32_create_random_data.txt'
```

Output of the plot:

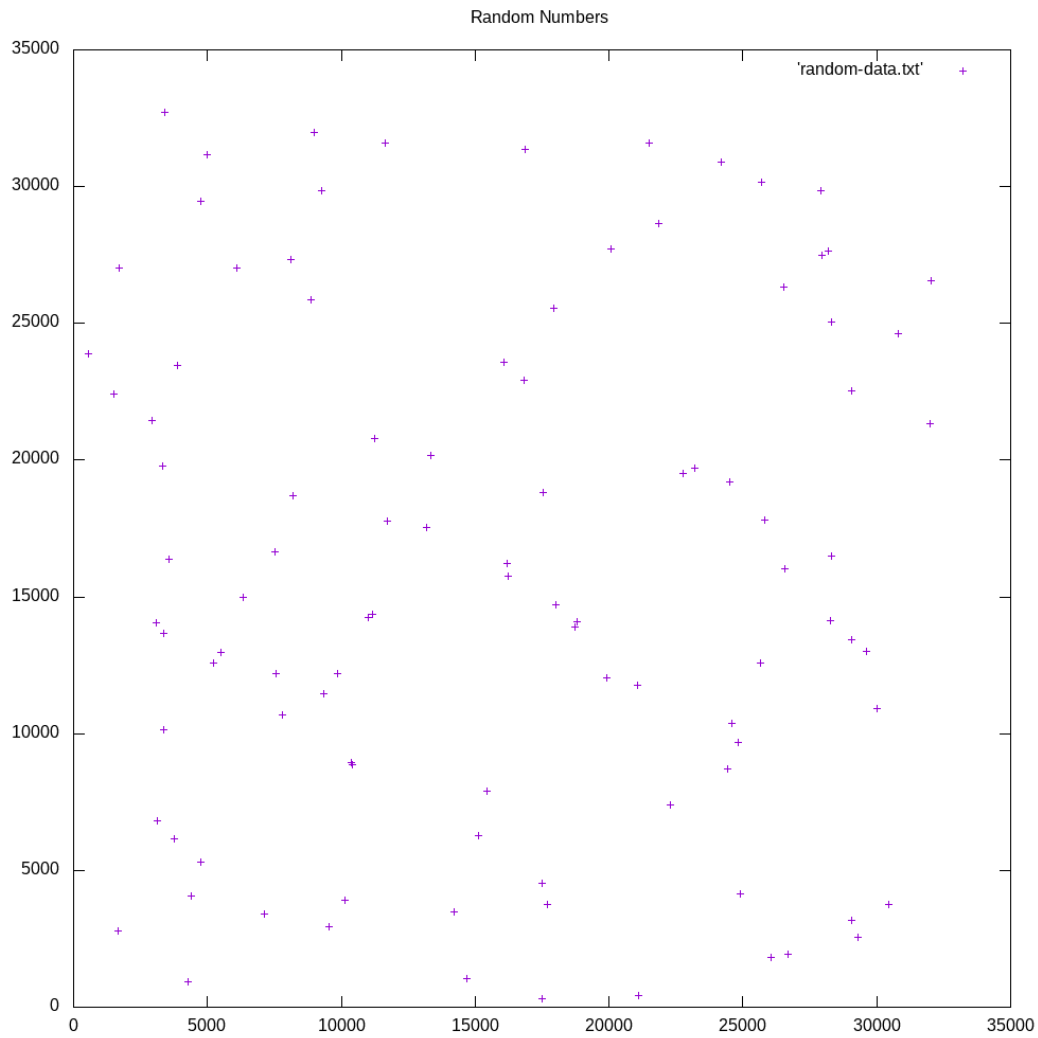


Figure 1: Output of the plot from `task32`

33. Let's plot some real data points

Contents of the `task33_plot_real_data.sh` file:

```
#!/bin/bash

dirs=`find lost24/monitor/ -type d -name "2011.11.*"`
dirs=`echo $dirs tr " " "\n" sort -u`

output_file="task33_plot_real_data.txt"

rm -f $output_file

for day in $dirs
do
```

```

max_temp_file=""
max_temp=0

for temp_file in `find $day -type f -name "*temps.txt"`
do
    temp=`grep "PROCESSOR_ZONE *[0-9][0-9]C" $temp_file -s cut -b 32-33`
    # echo $temp_file $temp
    if [ $temp -gt $max_temp ]
    then
        max_temp=$temp
        max_temp_file=$temp_file
    fi
done

day=${day##*/}
day=${day: -2}

echo $day : $max_temp \ $max_temp_file\
echo $day $max_temp >> $output_file

done

cat $output_file
gnuplot task33_plot_real_data.p

```

Contents of the task33_plot_real_data.txt file:

```

01 28
02 28
03 29
04 27
05 25
06 24
07 25
08 27
09 23
10 22
11 24
12 22
13 24
14 25
15 20
16 22
17 22
18 22
19 23
20 17
21 22
22 22
23 23
24 25
25 25
26 25
27 26
28 21

```

29 23
30 24

Contents of the task33_plot_real_data.p file:

```
set term postscript eps color blacktext 'Helvetica' 24
set output 'task33_plot_real_data.eps'

set title 'Max temperatures from November 2011'
set xlabel 'Day of the month'
set ylabel 'Temperature in Celsius'

set style line 2 lt 1 lw 2 pt 1 linecolor 1

plot 'task33_plot_real_data.txt' u 1:2 t "Temperatures" pt 2 ps .1 with lines
```

The following figure is the output of the execution with the command:

```
./task33_plot_real_data.sh
```

Contents of the task33_plot_real_data.eps file, output of the execution:

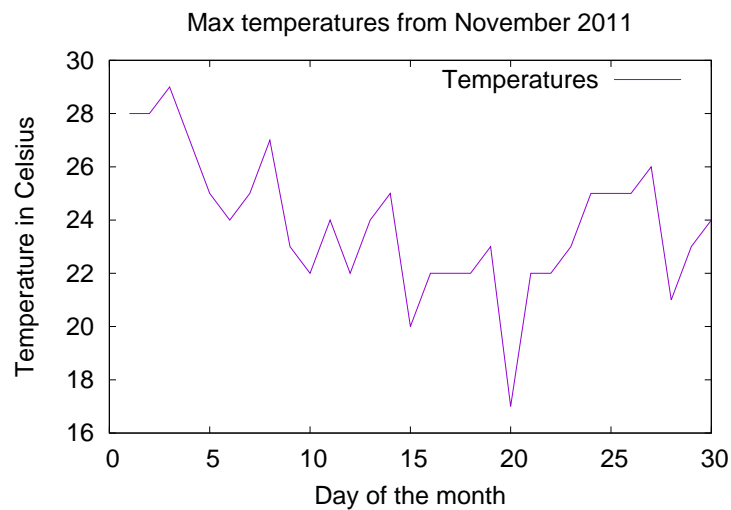


Figure 2: Contents of the task33_plot_real_data.eps file

34. Let's put some context

Contents of the task34_plot_min-max-temps-2011-11.sh file:

```
#!/bin/bash

dirs=`find lost24/monitor/ -type d -name "2011.11.*"`
dirs=`echo $dirs tr " " "\n" sort -u`

output_file="task34_plot_min-max-temps-2011-11.txt"
```

```

rm -f $output_file

for day in $dirs
do

    max_temp_file=""
    max_temp=0

    min_temp_file=""
    min_temp=99

    for temp_file in `find $day -type f -name "*temps.txt"`
    do
        temp=`grep "PROCESSOR_ZONE *[0-9][0-9]C" $temp_file -s cut -b 32-33`
        # echo $temp_file $temp
        if [ $temp -gt $max_temp ]
        then
            max_temp=$temp
            max_temp_file=$temp_file
        fi
        if [ $temp -lt $min_temp ]
        then
            min_temp=$temp
            min_temp_file=$temp_file
        fi
    done

    day=${day##*/}
    day=${day: -2}

    echo MAX TEMP $day : $max_temp \ $max_temp_file\
    echo MIN TEMP $day : $min_temp \ $min_temp_file\
    echo $day $max_temp $min_temp >> $output_file
    echo

done

# cat $output_file
gnuplot task34_plot_min-max-temps-2011-11.p

```

Contents of the task34_plot_min-max-temps-2011-11.txt file:

```

01 28 22
02 28 25
03 29 26
04 27 22
05 25 22
06 24 21
07 25 21
08 27 22
09 23 14
10 22 14
11 24 19
12 22 18
13 24 20
14 25 19

```

```
15 20 15
16 22 15
17 22 21
18 22 20
19 23 15
20 17 11
21 22 11
22 22 18
23 23 18
24 25 20
25 25 19
26 25 18
27 26 20
28 21 18
29 23 16
30 24 20
```

Contents of the task34_plot_min-max-temps-2011-11.p file:

```
set term postscript eps color blacktext 'Helvetica' 24
set output 'task34_plot_min-max-temps-2011-11.eps'

set title 'MAX and min temperatures from November 2011'
set xlabel 'Day of the month'
set ylabel 'Temperature in Celsius'

set style line 2 lt 1 lw 2 pt 1 linecolor 1

plot 'task34_plot_min-max-temps-2011-11.txt' u 1:2 t "MAX Temperatures" pt 2 ps .1 with
↵ lines, \
    'task34_plot_min-max-temps-2011-11.txt' u 1:3 t "MIN Temperatures" pt 2 ps .1 with
↵ lines
```

The following figure is the output of the execution with the command:

```
./task34_plot_min-max-temps-2011-11.sh
```

Contents of the task34_plot_min-max-temps-2011-11.eps file, output of the execution:

35. Let's generalize

Contents of the task35_plot_min-max-tempsgeneralized.sh file:

```
#!/bin/bash

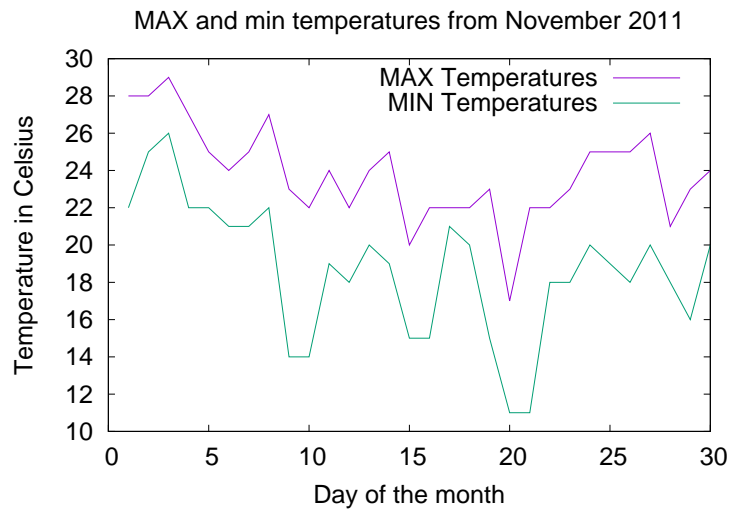
input_dir=$1

# Example input lost24/monitor/2011.10
dirs=`find ./ -type d -maxdepth 3 -wholename "$input_dir*" 2>/dev/null`
dirs=`echo $dirs tr " " "\n" sort -u`

output_file="task35_plot_min-max-temps_generalized.txt"

rm -f $output_file

for day in $dirs
```



```
do

max_temp_file=""
max_temp=0

min_temp_file=""
min_temp=99

for temp_file in `find $day -type f -name "*temps.txt"`
do
    temp=`grep "PROCESSOR_ZONE *[0-9][0-9]C" $temp_file -s cut -b 32-33`
    # echo $temp_file $temp
    if [ $temp -gt $max_temp ]
    then
        max_temp=$temp
        max_temp_file=$temp_file
    fi
    if [ $temp -lt $min_temp ]
    then
        min_temp=$temp
        min_temp_file=$temp_file
    fi
fi
done

day=${day##*/}
day=${day: -2}

echo MAX TEMP $day : $max_temp \ $max_temp_file\
echo MIN TEMP $day : $min_temp \ $min_temp_file\
echo $day $max_temp $min_temp >> $output_file
echo

done

# cat $output_file
gnuplot task35_plot_min-max-temps_generalized.p
```


Contents of the task35_plot_min-max-tempsgeneralized.txt file:

```
01 28 22
02 28 25
03 29 26
04 27 22
05 25 22
06 24 21
07 25 21
08 27 22
09 23 14
10 22 14
11 24 19
12 22 18
13 24 20
14 25 19
15 20 15
16 22 15
17 22 21
18 22 20
19 23 15
20 17 11
21 22 11
22 22 18
23 23 18
24 25 20
25 25 19
26 25 18
27 26 20
28 21 18
29 23 16
30 24 20
```

Contents of the task35_plot_min-max-tempsgeneralized.p file:

```
set term postscript eps color blacktext 'Helvetica' 24
set output 'task35_plot_min-max-tempsgeneralized.eps'

set title 'MAX and min temperatures for month in input'
set xlabel 'Day of the month'
set ylabel 'Temperature in Celsius'

set style line 2 lt 1 lw 2 pt 1 linecolor 1

plot 'task35_plot_min-max-tempsgeneralized.txt' u 1:2 t "MAX Temperatures" pt 2 ps .1 with
↵ lines, \
    'task35_plot_min-max-tempsgeneralized.txt' u 1:3 t "MIN Temperatures" pt 2 ps .1 with
↵ lines
```

The following figure is the output of the execution with the command:

```
./task35_plot_min-max-tempsgeneralized.sh lost24/monitor/2011.11
```

36. Let's make more refined commands

Contents of the task36_min-max-tempsgetops_wrapper.sh file:

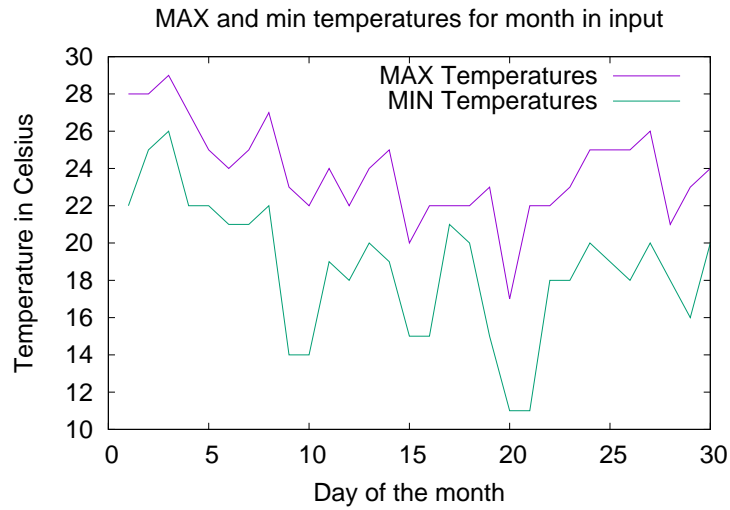


Figure 3: Contents of the task35_plot_min-max-temps_generalized.eps file

```
#!/bin/bash

if [ "$#" -lt 1 ]
then
    echo ""
    echo "You have passed no argumets. Try again, but follow the rules!"
    echo ""
    ./task36_min-max-temps_getopts_wrapper.sh -h
    exit 1
fi

c=false
w=false
ascii=false

while getopts ":cwbah" option
do

    case $option in

        ## COLDEST TEMPERATURES ##
        c          continue
                ;;

        ## WARMEST TEMPERATURES ##
        w          continue
                ;;

        ## BOTH TEMPERATURES ##
        b          w=true
                continue
                ;;

        ## ASCII output ##
    esac
done
```

```

a            continue
;;

## HELP ##
h            echo "-- HELP --"
            echo "-c: coldest teperatures"
            echo "-w: warmest teperatures "
            echo "-b: both of the above"
            echo "-a: ASII output"
            echo "-h: help this menu, exits the program even if there are other arguments
            ↪ present"
            echo ""
            echo "After these arguments pass the date of the folder you want to search for"
            echo ""
            exit 0
;;

## ANYTHING ELSE ##
*            exit 1
;;

esac
done

if [ "$c" == false ] && [ "$w" == false ]
then
    ./task36_min-max-temps_getops_wrapper.sh -h
    exit 1
fi

input_dir=$2

# Example input lost24/monitor/2011.10
dirs=`find ./ -type d -maxdepth 3 -wholename "$input_dir*" 2>/dev/null`
dirs=`echo $dirs tr " " "\n" sort -u`

output_file="task36_min-max-temps_getops_wrapper.txt"

rm -f $output_file

for day in $dirs
do

    max_temp_file=""
    max_temp=0

    min_temp_file=""
    min_temp=99

    for temp_file in `find $day -type f -name "*temps.txt"`
    do
        temp=`grep "PROCESSOR_ZONE *[0-9][0-9]C" $temp_file -s cut -b 32-33`
        # echo $temp_file $temp

        # If the user wants the warmest temperatures then we calculate them
        if [ "$w" == true ]
        then
            if [ $temp -gt $max_temp ]
            then

```

```

        max_temp=$temp
        max_temp_file=$temp_file
    fi
fi

# If the user wants the coldest temperatures then we calculate them
if [ "$c" == true ]
then
    if [ $temp -lt $min_temp ]
    then
        min_temp=$temp
        min_temp_file=$temp_file
    fi
fi
done

day=${day##*/}
day=${day: -2}

echo -n $day " " >> $output_file

if [ "$w" == true ] ; then echo MAX TEMP $day : $max_temp \ $max_temp_file\ ; echo -n
↪ $max_temp " " >> $output_file ; fi
if [ "$c" == true ] ; then echo MIN TEMP $day : $min_temp \ $min_temp_file\ ; echo -n
↪ $min_temp " " >> $output_file ; fi

echo " " >> $output_file

echo ""

done

gnuplot_output_file=${input_dir: -7}
gnuplot_output_file=`echo "$gnuplot_output_file" sed "s/\./_/g"`

# cat $output_file
gnuplot -e "max='$w'" \
-e "min='$c'" \
-e "ascii='$ascii'" \
-e "output_file='$gnuplot_output_file'" \
task36_min-max-temps_getops_wrapper.p

```

Contents of the task36_min-max-temps_getops_wrapper.p file:

```

if ascii eq 'true' {

    set terminal dumb
    set table "task36_min-max-temps_getops_wrapper_ascii_".output_file.".txt"

} else {

    set term postscript eps color blacktext 'Helvetica' 24
    set output "task36_min-max-temps_getops_wrapper_".output_file.".eps"

    set xlabel 'Day of the month'
    set ylabel 'Temperature in Celsius'

```

```

    set style line 2 lt 1 lw 2 pt 1 linecolor 1
}

if max eq 'true' && min eq 'true' {
    set title 'MAX and min temperatures for month in input'
    plot 'task36_min-max-temps_getops_wrapper.txt' u 1:2 t "MAX Temperatures" pt 2 ps .1
    ↪ with lines, \
        'task36_min-max-temps_getops_wrapper.txt' u 1:3 t "MIN Temperatures" pt 2 ps .1
    ↪ with lines ;
} else {
    if min eq 'true'{
        set title 'min temperatures for month in input'
        plot 'task36_min-max-temps_getops_wrapper.txt' u 1:2 t "min Temperatures" pt 2 ps
        ↪ .1 with lines ;
    }
    if max eq 'true'{
        set title 'MAX temperatures for month in input'
        plot 'task36_min-max-temps_getops_wrapper.txt' u 1:2 t "MAX Temperatures" pt 2 ps
        ↪ .1 with lines ;
    }
}

if ascii eq 'true'{
    unset table
}

```

Contents of the task36_min-max-temps_getops_wrapper.txt file:

```

01 28 22
02 28 25
03 29 26
04 27 22
05 25 22
06 24 21
07 25 21
08 27 22
09 23 14
10 22 14
11 24 19
12 22 18
13 24 20
14 25 19
15 20 15
16 22 15
17 22 21
18 22 20
19 23 15
20 17 11
21 22 11
22 22 18
23 23 18
24 25 20
25 25 19
26 25 18
27 26 20

```

28 21 18
29 23 16
30 24 20

The following is the output of the execution with the command:

```
./task36_min-max-temps_getops_wrapper.sh -b lost24/monitor/2011.11
```

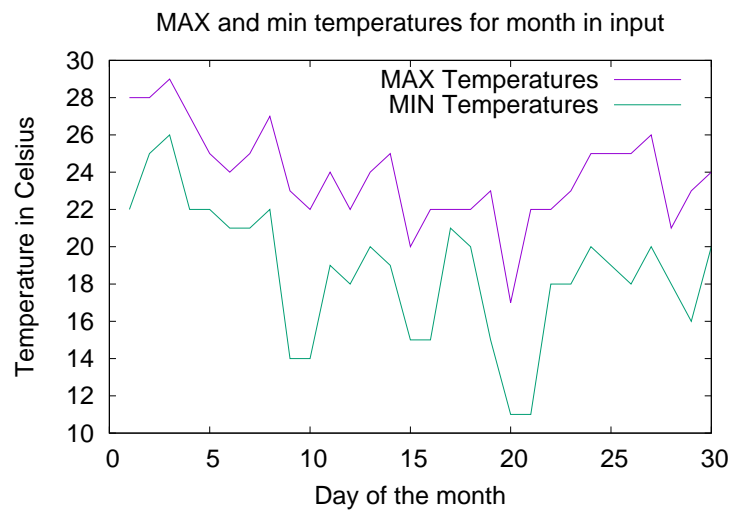


Figure 4: Contents of the task36_min-max-temps_getops_wrapper_2011_11.eps file

Contents of the task36_min-max-temps_getops_wrapper_ascii_2011_11.txt file:

```
# Curve 0 of 2, 30 points
# Curve title: "MAX Temperatures"
# x y type
1 28 i
2 28 i
3 29 i
4 27 i
5 25 i
6 24 i
7 25 i
8 27 i
9 23 i
10 22 i
11 24 i
12 22 i
13 24 i
14 25 i
15 20 i
16 22 i
17 22 i
18 22 i
19 23 i
20 17 i
21 22 i
```

```
22 22 i
23 23 i
24 25 i
25 25 i
26 25 i
27 26 i
28 21 i
29 23 i
30 24 i
```

```
# Curve 1 of 2, 30 points
# Curve title: "MIN Temperatures"
```

```
# x y type
1 22 i
2 25 i
3 26 i
4 22 i
5 22 i
6 21 i
7 21 i
8 22 i
9 14 i
10 14 i
11 19 i
12 18 i
13 20 i
14 19 i
15 15 i
16 15 i
17 21 i
18 20 i
19 15 i
20 11 i
21 11 i
22 18 i
23 18 i
24 20 i
25 19 i
26 18 i
27 20 i
28 18 i
29 16 i
30 20 i
```
