Basic of Informatics – Lab. 5.

1. **Unix - scripting**

- Create file *test1* and define variables:

FILE1=~/FILENAME1 ; FILE2=~/FILENAME2 -filename1 and filename2 names of existing files in Your working directory

file1_name=$(stat –c%n $FILE1) (the same for FILE2)

file1_date=$(stat –c%Y $FILE1) (the same for FILE2)

if test *$file1_date –gt $file2_date* **then** INSTRUCTION1 **else** INSTRUCTION2 **fi**

(instruction1 and instruction2 yours own instructions, echo?)

- Execute file *test1* with: **./test1**
- Create a file: *bashrc_my* with content:

```
FILE=~/.bashrc
if [ -e $FILE ]
then
if [ -f $FILE ]
 then
 echo "$FILE is a regular file."
fi
if [ -d $FILE ]
then
 echo "$FILE is a directory."
fi
if [ -r $FILE ]
then echo "$FILE is readable."
fi
if [ -w $FILE ]
then echo "$FILE is writable."
fi
else echo "$FILE does not exist"
exit 1
 fi
 exit
```

2. if, elif, else

- Create file: *test2* with content:

answer=yes

if [$answer ==”yes”]; then

echo “The answer is YES”

elif [$answer==”no”]; then

echo “The answer is NO. “

else

echo “The answer is UNKOWN. “

fi

- Define variable answer as: yes, no, default text and execute a file *test1*

3. **Case statement**

- Type into case1 file1
- : *echo –n “Which number?”* then use command: *read Number*

```
case $Number in
1)   echo “One! ;;
2)   echo “Two” ;;
3)   echo “etc” ;;

*) echo “Invalid number” ;;

esac
```

4. String and integer expressions, reading input (create a files and execute them)

- Create new file: integers and fill it with content:

```
INT=-5
if [ -z $INT ];
 then
echo "$INT is empty." >&2
exit 1
fi
if [ $INT -eq 0 ];
then
echo "$INT is zero."
else
if [ $INT -lt 0 ];
then
echo "$INT is negative."
else
echo "$INT is positive."
fi
if [ $((INT % 2)) -eq 0 ];
 then
echo "$INT is even."
else
echo "$INT is odd."
fi
```

fi

- Create a file *read1* with content:

```
echo -n "Please enter an integer -> "
read int

if [ -z $int ];
then
echo "int is empty." >&2
exit 1
fi
if [ $int -eq 0 ];
 then
echo "int is zero."
else
if [ $int -lt 0 ];
 then
echo "int is negative."
else
echo "int is positive."
fi
if [ $((int % 2)) -eq 0 ];
 then
echo "int is even."
else
echo "int is odd."
fi
fi
```

- Create a file: *menu*

```
clear
echo "
PleaseSelect:
1. Display System Information
2. Display Disk Space
3. Display Home Space Utilization
0. Quit
 "
 read -p "Enter selection [0-3] > "
 if [[ $REPLY =~ ^[0-3]$ ]];
 then
 if [ $REPLY == 0 ];
 then
 echo "Program terminated."
 exit
 fi
 if [ $REPLY == 1 ];
 then
 echo "Hostname: $HOSTNAME"
 uptime
 elif [ $REPLY == 2 ];
 then
 df -h
```

elif [$REPLY == 3];
then
echo "Home Space Utilization ($USER)"
du -sh $HOME
fi
else
echo "Invalid entry." >&2
exit 1
fi

5. **Loops: while, until, for**

- Create a file *while:*
  cnt=1
  while [$cnt -le 4]; do
  echo $cnt
  cnt=$((cnt + 1))
  done
  echo "Finished."
- Create a file *until:*
  cnt=1
  until [$cnt -gt 4]; do
  echo $cnt
  cnt=$((cnt + 1))
  done
  echo "Finished."
- Create a file GPS.txt with content:
  Cracow 50.123, 19.234
  Warsaw 45.123, 18.123
  Then create a file GPS!:
  while read location length width; do
  *printf "Location: %s\tLength: %s\tWidth: %s\n" *
  *$location *
  *$length *
  $width
  done < GPS.txt
- Create a file *for*
  for i in {M..Z}; do
  echo $i;
  done
- Create a file *for1*
  for i in 2.0 2.5 3.0 3.5; do
  echo $i;
  done
- Create a file *for2*
  for ((i=0; i<10; i=i+1)); do
  echo $i
  done
  Execute all the files created in this exercise