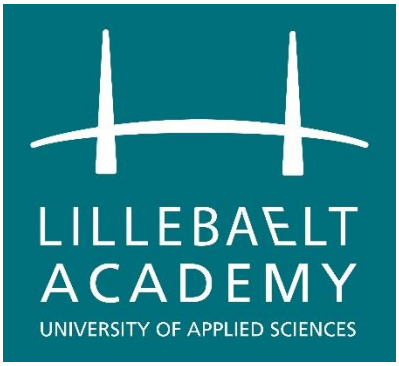
IT Technology
LIN 13 Linux
File statistics and
HTML reporting script.
2017.06.11



LILLEBAELT ACADEMY
UNIVERSITY OF APPLIED SCIENCE

Authors:
Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

Sunday 11 June 2017

Table of Contents

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

Introduction

This document shows basic file statistic implementation in an HTMLreporting script. The audience of this document are the teachers and any of our classmates who is interested in to check. The audience must have done and understand chapter 18 and 19 in ” THE LINUX COMMAND LINE. by William E. Shotts ”

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

LIN 13 File statistics and HTML reporting script.

```
# test-file: Evaluate the status of a
file 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 "$FILE is writable."
    fi
    if [ -x "$FILE" ]; then
        echo "$FILE is executable/searchable."
    fi
else
    echo "$FILE does not
    exist" exit 1
fi
exit
```

Merge the above program with what was learned about the HTML report. Make a report for a given file on the system. It is out of scope here to be able to choose or input which file to investigate. This would e.g. require the involvement of a web server and a cgi script. E.g. php or Pyhton.

Let’s create a file what will we use in the web browser later on:

root@ubuntu:/bin# nano assignment.sh

In edit mode we add the basic HTML structure and then change a bit:

```
#!/bin/bash
# Program to output system's information page

myTITLE="System information Report from $HOSTNAME"

echo "<HTML>
            <HEAD>
                  <TITLE>$myTITLE</TITLE>

            </HEAD>
            <BODY>
                  <H1>$myTITLE</H1>
            </BODY>
      </HTML>"
```

Furthermore, lets copy and past the program above the example:

```
#!/bin/bash
# Program to output system's information page

myTITLE="System information Report from $HOSTNAME"

echo "<HTML>
            <HEAD>
                  <TITLE>$myTITLE</TITLE>

            </HEAD>
            <BODY>
                  <H1>$myTITLE</H1>
            </BODY>
      </HTML>"
# test-file: Evaluate the status of a file
FILE=/bin/assignment.sh

if [ -e "$FILE" ]; then

if [ -f "$FILE" ]; then
echo "$FILE is a regular file.<br>"

fi
if [ -d "$FILE" ]; then

echo "$FILE is a directory.<br>"

fi
if [ -r "$FILE" ]; then

echo "$FILE is readable.<br>"
fi
if [ -w "$FILE" ]; then

echo "$FILE is writable.<br>"
fi

if [ -x "$FILE" ]; then
echo "$FILE is executable/searchable.<br>"
fi

else

echo "$FILE does not exist" exit 1

fi
exit
```

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

The HTML code can be integrated in a bash script. We started with that at the 1st line:

#!/bin/bash

We declared a variable:

myTITLE="System Information Report $HOSTNAME"

Please note that spaces are not allowed here.

We added
 because we want the truth statuses to follow each other in new lines.

Then redirect the output of the script to a file and point your browser to the file:

assignment.sh < assignment.html

The content of the html file will be then:

```
<HTML>
        <HEAD>
                <TITLE>System information Report from ubuntu</TITLE>

        </HEAD>
        <BODY>
              <H1>System information Report from ubuntu</H1>
        </BODY>
    </HTML>
/bin/assignment.sh is a regular file.<br>
/bin/assignment.sh is readable.<br>
/bin/assignment.sh is writable.<br>
/bin/assignment.sh is executable/searchable.<br>
```

In the browser it looks like this:

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

To be able to add the output from the date command as a variable we can use tha (`) back quotation mark:

root@ubuntu:/bin# d=`date`

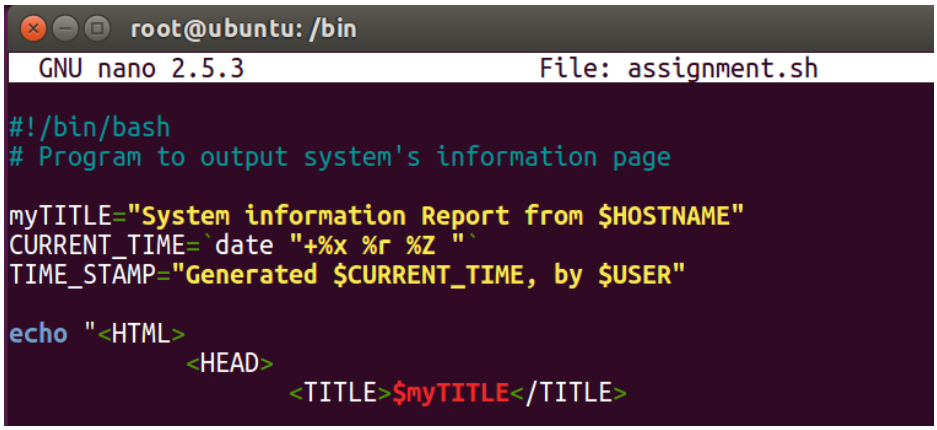
root@ubuntu:/bin# echo $d

Sat Jun 10 04:05:36 PDT 2017

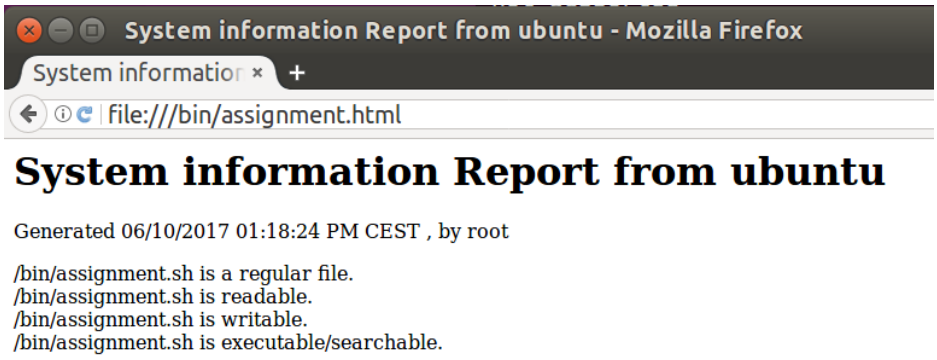
Note that spaces are not allowed:

root@ubuntu:/bin# d=`date "+%m"`

Then add the date to the system report:



Refresh the page in web browser and note the result:



As we can see the System Report shows the Hostname, we see the exact date, time and then which mode were we when executed the command. Furthermore we can see the file is a regular one, it's readable, writable and executable/searchable.

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk

Conclusion

When we went in details of the exercise it was really useful, and learn a lot about HTML implementation.

Nikolay Petrov:niko010h@edu.eal.dk
Milan Vince:mila1025@edu.eal.dk
Dilshad Khalaf: dils0008@edu.eal.dk