CST 334 (Operating Systems)

Dr. Glenn Bruns

# Lab: Intermediate Bash -- regular expressions

There are hints below but refer to them only when you really need to.

1. Create a file 'movies.txt' containing the following lines:

My ratings:

Exit through the gift shop \*\*\*\*

Star Wars \*\*

Headhunters \*\*\*

$ \*

1. Run the following command and explain the output: grep $ movies.txt
2. Run the following command and explain the output: grep '$' movies.txt
3. Run the following command and explain the output: grep '\$' movies.txt
4. Run the following command and explain the output: grep \* movies.txt
5. Run the following command and explain the output: grep '\*\*\*\*' movies.txt
6. Run the following command and explain the output: grep '\\*\\*\\*\\*' moves.txt
7. For a fun change of pace, let's work on regular expressions using an online tool. Go to the following page and start drilling on regular expressions:

[regexone.com/lesson/introduction\_abcs](https://regexone.com/lesson/introduction_abcs)

I suggest skipping problems 11-13 on match groups. Go back to them if you finish the other problems. Match groups are useful when using regular expressions in an editor, or with sed (a tool we'll discuss soon).

1. Does your text editor support regular expressions? Find out, and if it does, try doing some of the exercises in part 1 in a text document.

## Hints

1. -
2. Here $ is interpreted as a bash special character. Remember that $ is used in front of a variable when we want the value of the variable. In this case there is no variable name after $, so $ is treated as the empty string, and matches everything.
3. '$' is not interpreted by bash, but is interpreted by grep as the regular expression that means 'end of line'.
4. We want to match on lines containing $. We use single quotes so bash won't interpret it, and a \ so that grep won't interpret it!
5. Bash interprets the \* as 'any string', so weird stuff happens.
6. I'm not sure what \*\*\*\* means as a regular expression, since \* is only meaningful if preceded by another character.
7. Now we are searching for lines with four consecutive \* characters.
8. -
9. -