UNIX and Shell Programming

Assignment 7: sed

20 August 2019

1 Exercise 1

1. Launch a terminal.
2. Create and save the following file. Call it `a7-e1-f1`. Note that the fields are separated by one or more spaces (randomly). The first field is the first name, the second field is the last name, and the third field is the age.

```
John Adams 55
George Bull 77
Anne Blue 99
Janet Blue 67
Ben Benjamin 78
Ted White 32
```

3. Use a one-line sed command to reorganize the file using the comma/tab pattern shown in the following file. Note that the last name is before the first name, and there is only one space between the names and a space before the numbers. Use the same name for the new file.

```
Adams, John 55
Bull, George 77
Blue, Anne 99
Blue, Janet 67
Benjamin, Ben 78
White, Ted 32
```

4. Sort the file first according to the last name and then according to the age. Watch out for the comma after the last name. Use the same name for the new file.
5. Quit the terminal.

2 Exercise 2

1. Launch a terminal.
2. Create and save the following file. Call it `a7-e2-f1`.

```
UNIX is as UNIX does.
And D03 is as DOS does.
But UNIX is not as DOS does.
Nor is DOS as UNIX does.
So, if UNIX was as DOS does,
Would UNIX be DOS
Or would DOS be UNIX?
Or to put the question another way:
Is an operating system by any other name
As beautiful as a UNIX operating system?
```

3. Use a sed script (`a7-e2-f2.sed`) and a sed command to put a set of five asterisks at the beginning and end of each line that contains the pattern UNIX. Call the new file `a7-e2-f3`.
4. Quit the terminal.

3 Exercise 3

1. Launch a terminal.
2. Create a file of fifty lines (each line can have only a word or two). Call the file `a7-e3-f1`.
3. Use the cat command to insert a line number at the beginning of each line in a file. This will help to check the result of the next steps.
4. Write a sed script (`a7-e3-f2.sed`) and a sed command to split the file into four files. The first file, called `a7-e3-f3`, contains lines 10 to 15. The second file, called `a7-e3-f4`, contains lines 20 to 30. The third file, called `a7-e3-f5`, contains lines 31 to 37. The fourth file, called `a7-e3-f6`, contains the rest of the file.
5. Print all of the files created in this session and verify the output.
6. Quit the terminal.

4 Exercise 4

1. Launch a terminal.

2. Create the followingfile and call it `a7-e4-f1`. Each line in the file is an absolute pathname of a file.

```
bin/date
bin/programs/cal
usr/bin/date
usr/report/file1
usr/report/letters/lett1
/spool/mails
```

3. Write a sed script (`a7-e4-f2`) and a sed command to extract the lowest level directory and the name of the file from the path (separated by spaces) and store it in a file called `a7-e4-f3`. The file should look like the following (directory then file}:

```
/bin date
/bin/programs cal
/usr/bin date
/usr/report file1
/usr/report/letters lett1
/spool mails
```

4. Quit the terminal.

5 Exercise 5

1. Launch a terminal.
2. Create the following file and call it `a7-e5-f1`. The file is a C program that multiplies two numbers. It contains some comments, which begin with the two- character token `/*` and end with the two-character token `*/`. In this program, comments can be on one line or can span more than a line.

```
/* The greeting program. This program demonstrates */
/* some of the components of a simple C program. */
/* Written by: your name here ' */
/* Date: date program written */

#include <stdio.h>
int main (void)
{
/* Statements */
```

```
printf("Hello World!\n");

return 0;
} /* main */
```

3. Write a sed script `a7-e5-f2.sed` and a sed command to delete the comments from the file. Call the new file `a7-e5-f3`. You will need to pay special attention to the slashes and asterisks (they need to be quoted).

4. Quit the terminal.

6 Exercise 6

1. Launch a terminal.

2. Create the following file and call it `a7-e6-f1`. The file is a C program that multiplies two numbers. It contains some comments which begin with the two-character token `/*` and end with the two-character token `*/`. In this program, comments can be on one line or can span more than a line

```
/* This program reads two integer numbers from the
    keyboard and prints their product.
    Written by:
    Date:

*/

/* Statements */

scanf ("%d", &number1);
scanf ("%d", &number2);
result = number1 * number2;
printf ("%d", result);
return 0;
} /* main */
```

3. Write a sed script `a7-e6-f2.sed` and a sed command to delete the comments from the file. Call the new file `a7-e6-f3`.

4. Quit the terminal.