

Yandex

Unix Command Line

File content exploration

Part I

How can we explore files?

Files exploration commands

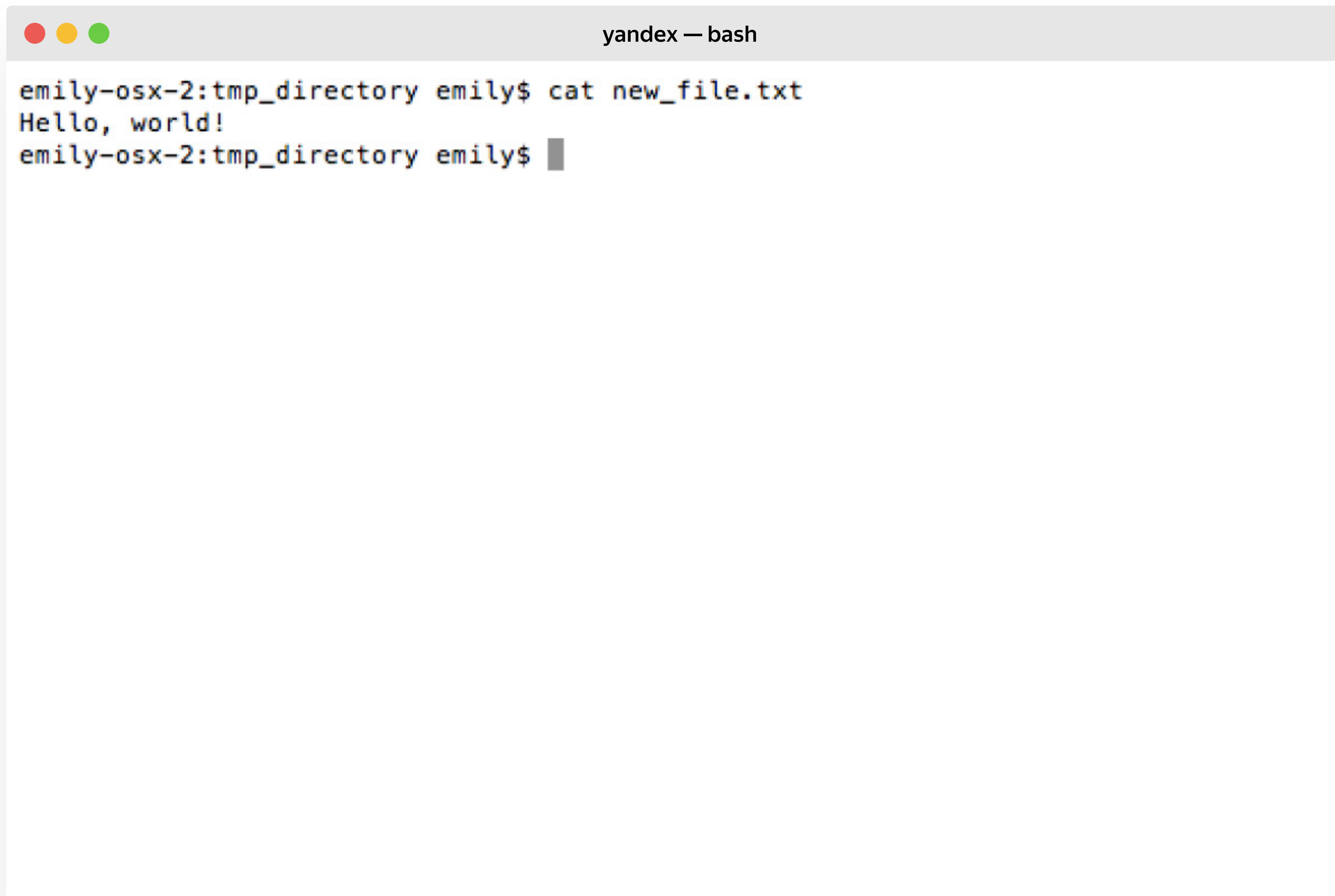
- › cat
- › head
- › tail
- › more
- › less
- › wc
- › grep

cat

- › displays the contents of a file at the command line
- › copies or append text files into a document

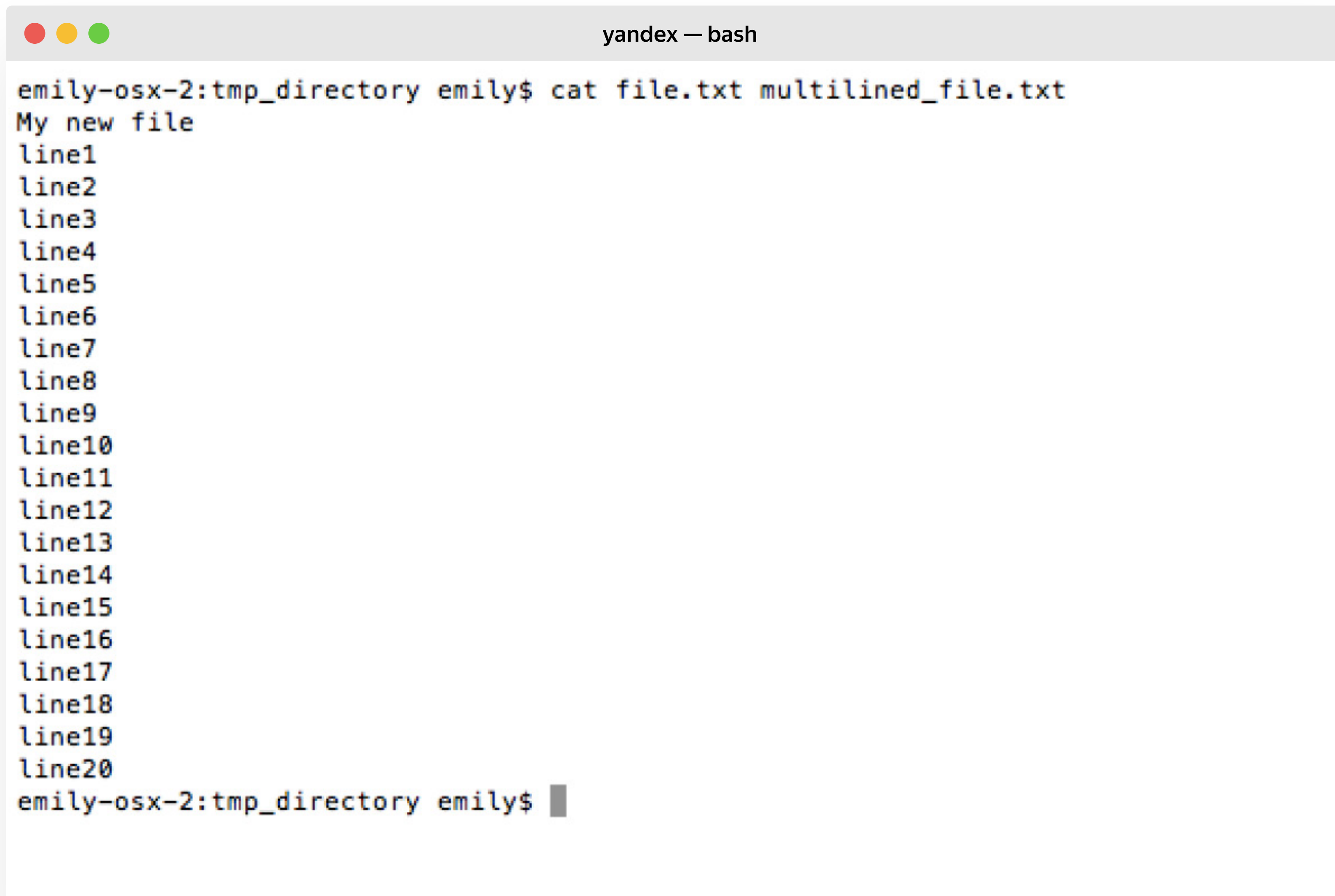
cat [OPTION]... [FILE]...

cat

A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$" followed by the command "cat new_file.txt". The output of the command is "Hello, world!". The prompt "emily-osx-2:tmp_directory emily\$" is shown again on the next line with a cursor.

```
yandex — bash  
emily-osx-2:tmp_directory emily$ cat new_file.txt  
Hello, world!  
emily-osx-2:tmp_directory emily$
```

cat

A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$" followed by the command "cat file.txt multilined_file.txt". The output of the command is displayed as a single block of text, starting with "My new file" and followed by 20 lines labeled "line1" through "line20". The prompt "emily-osx-2:tmp_directory emily\$" appears again at the bottom with a cursor.

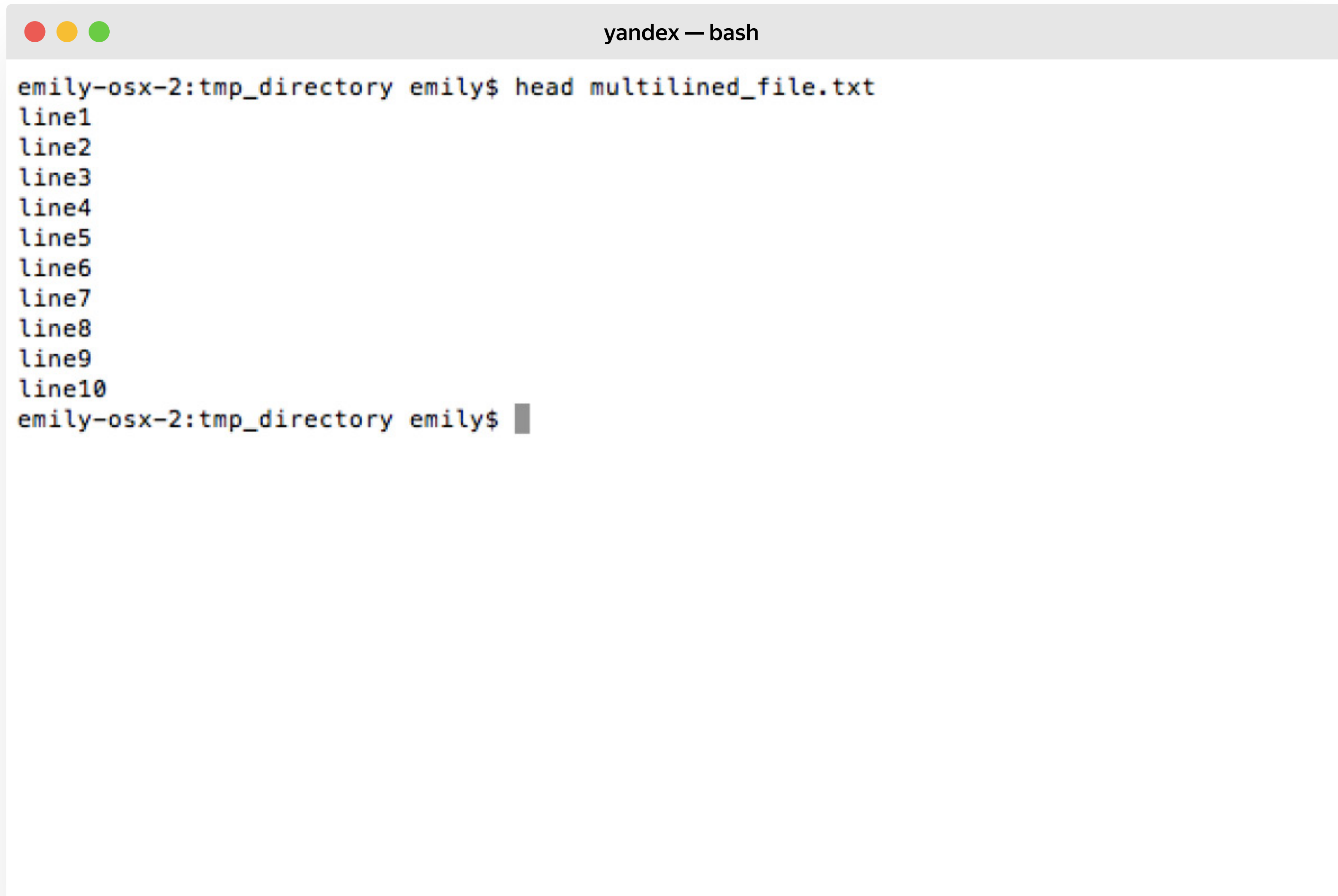
```
yandex — bash
emily-osx-2:tmp_directory emily$ cat file.txt multilined_file.txt
My new file
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
emily-osx-2:tmp_directory emily$
```

head

- › prints the first part of files

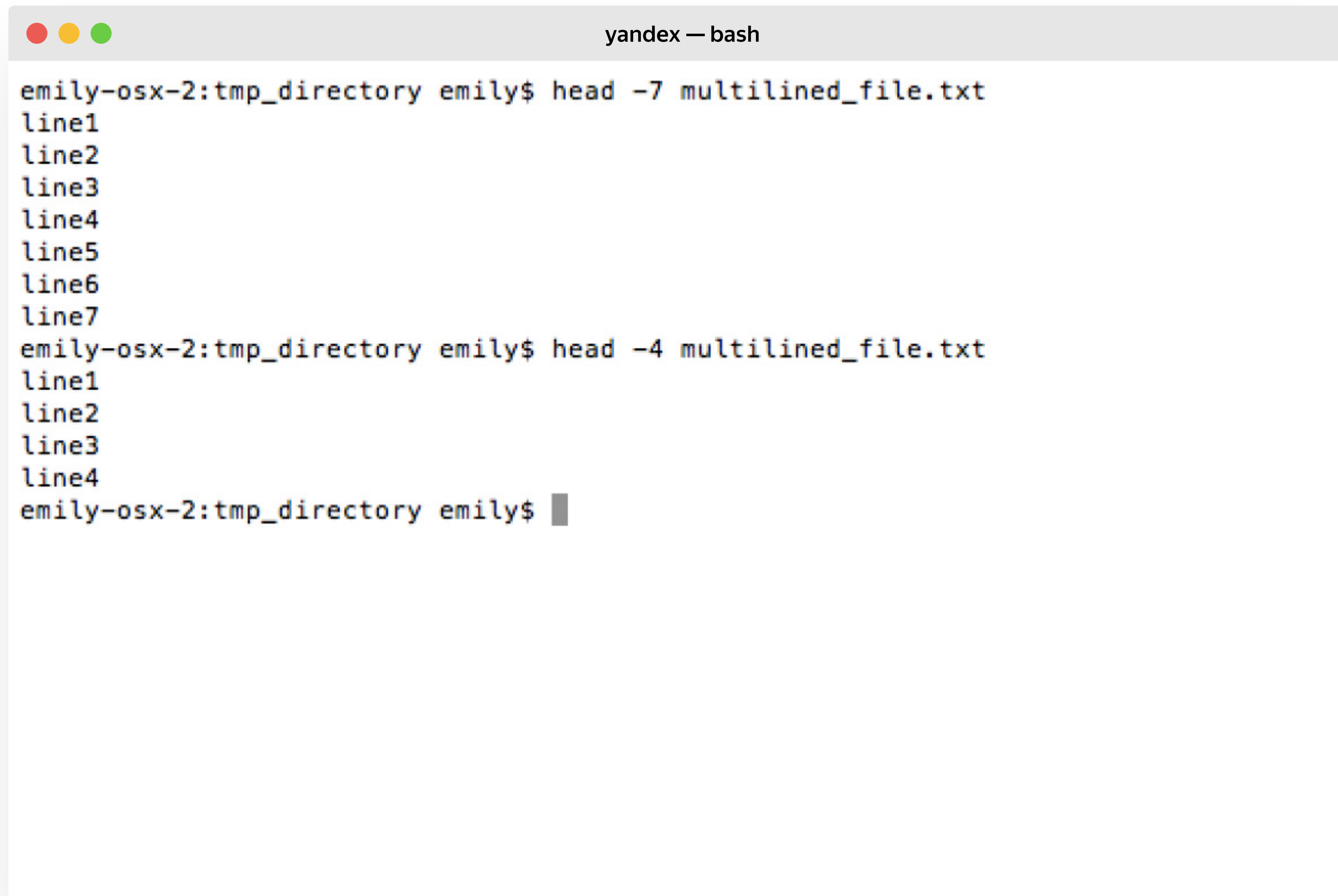
head [OPTION]... [FILE]...

head

A terminal window titled "yandex — bash" with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$" followed by the command "head multilined_file.txt". The output of the command is a list of ten lines, labeled "line1" through "line10", each on a new line. The prompt "emily-osx-2:tmp_directory emily\$" is shown again at the bottom of the terminal, followed by a cursor.

```
yandex — bash
emily-osx-2:tmp_directory emily$ head multilined_file.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
emily-osx-2:tmp_directory emily$
```

head -n



```
yandex — bash

emily-osx-2:tmp_directory emily$ head -7 multilined_file.txt
line1
line2
line3
line4
line5
line6
line7
emily-osx-2:tmp_directory emily$ head -4 multilined_file.txt
line1
line2
line3
line4
emily-osx-2:tmp_directory emily$
```

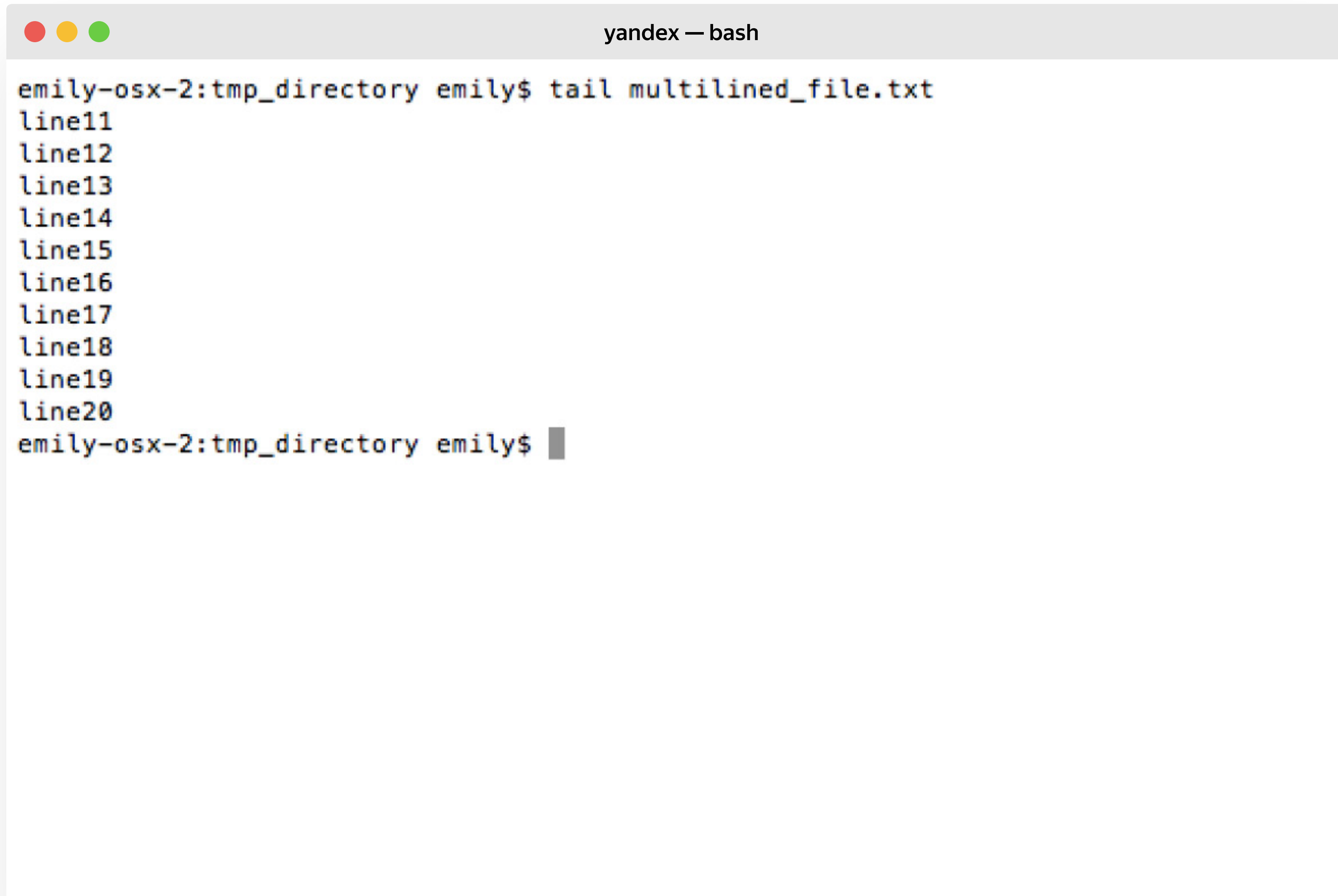
The image shows a terminal window titled "yandex — bash". The prompt is "emily-osx-2:tmp_directory emily\$". The first command executed is "head -7 multilined_file.txt", which outputs seven lines labeled "line1" through "line7". The second command is "head -4 multilined_file.txt", which outputs four lines labeled "line1" through "line4". The prompt "emily-osx-2:tmp_directory emily\$" is shown again at the end, with a cursor.

tail

- › prints the last part of files

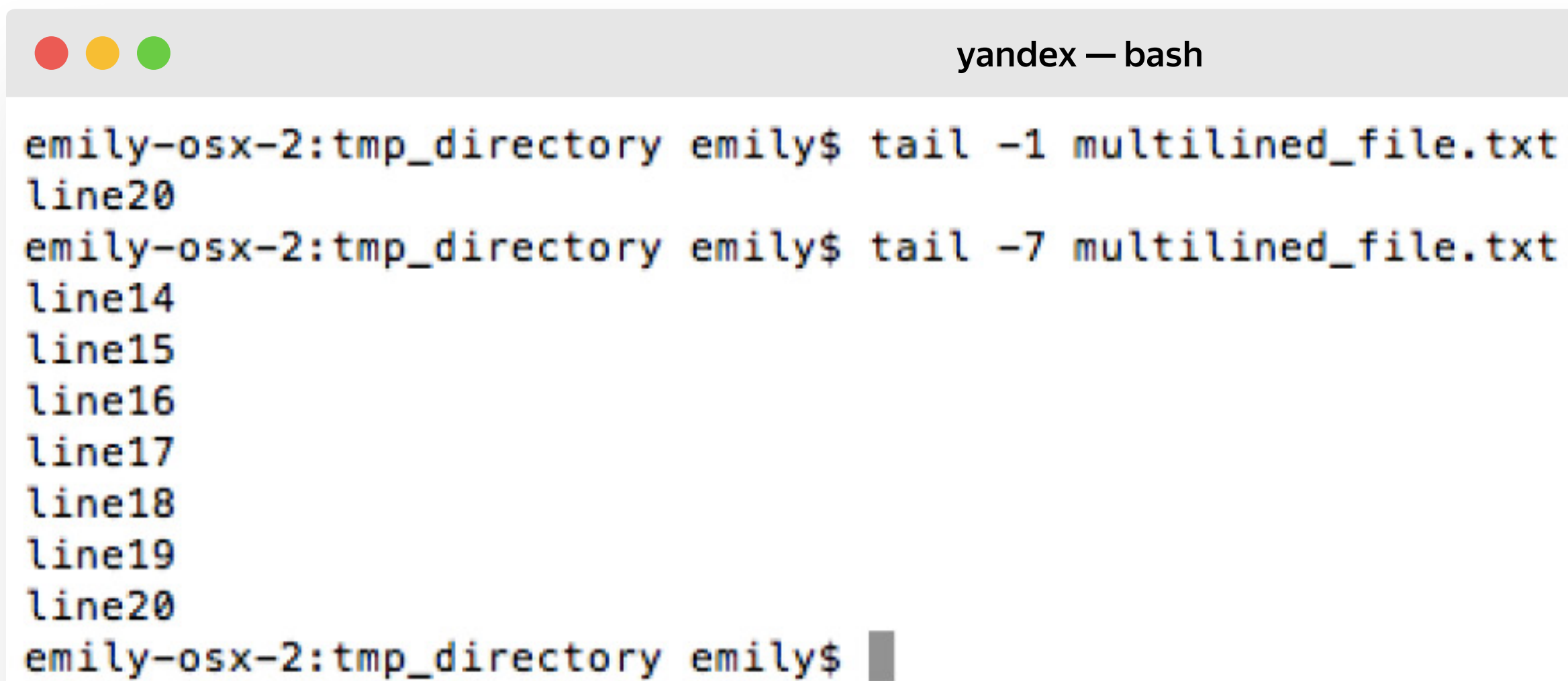
tail [OPTION]... [FILE]...

tail

A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$" followed by the command "tail multilined_file.txt". The output consists of ten lines labeled "line11" through "line20". The prompt "emily-osx-2:tmp_directory emily\$" is shown again at the bottom with a cursor.

```
yandex — bash
emily-osx-2:tmp_directory emily$ tail multilined_file.txt
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
emily-osx-2:tmp_directory emily$
```

tail -n

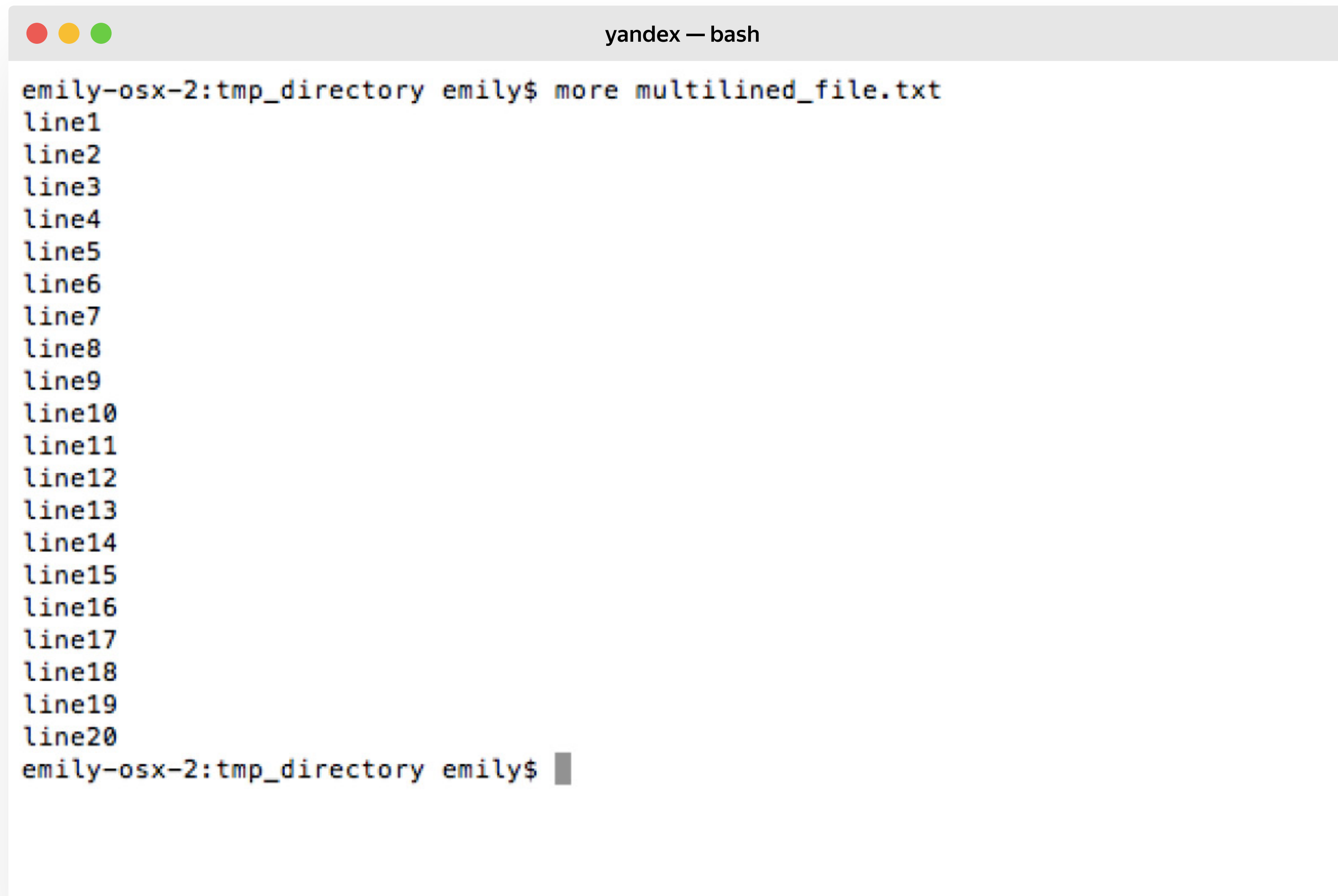
A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows two commands being executed. The first command, "tail -1 multilined_file.txt", outputs "line20". The second command, "tail -7 multilined_file.txt", outputs a list of seven lines: "line14", "line15", "line16", "line17", "line18", "line19", and "line20". The prompt "emily-osx-2:tmp_directory emily\$" is shown at the end of the output.

```
yandex — bash
emily-osx-2:tmp_directory emily$ tail -1 multilined_file.txt
line20
emily-osx-2:tmp_directory emily$ tail -7 multilined_file.txt
line14
line15
line16
line17
line18
line19
line20
emily-osx-2:tmp_directory emily$
```

more

- › displays text, one screen at a time

more

A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows a user at "emily-osx-2:tmp_directory emily\$" running the command "more multilined_file.txt". The output consists of 20 lines, each labeled from "line1" to "line20". The terminal cursor is positioned at the end of the 20th line, ready for the next command.

```
yandex — bash
emily-osx-2:tmp_directory emily$ more multilined_file.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
line20
emily-osx-2:tmp_directory emily$
```

less

- › displays text, allows scrolling

less



```
yandex — bash  
line1  
line2  
line3  
line4  
line5  
line6  
line7  
line8  
line9  
line10  
line11  
line12  
line13  
line14  
line15  
line16  
line17  
line18  
line19  
line20  
multilined_file.txt (END)
```

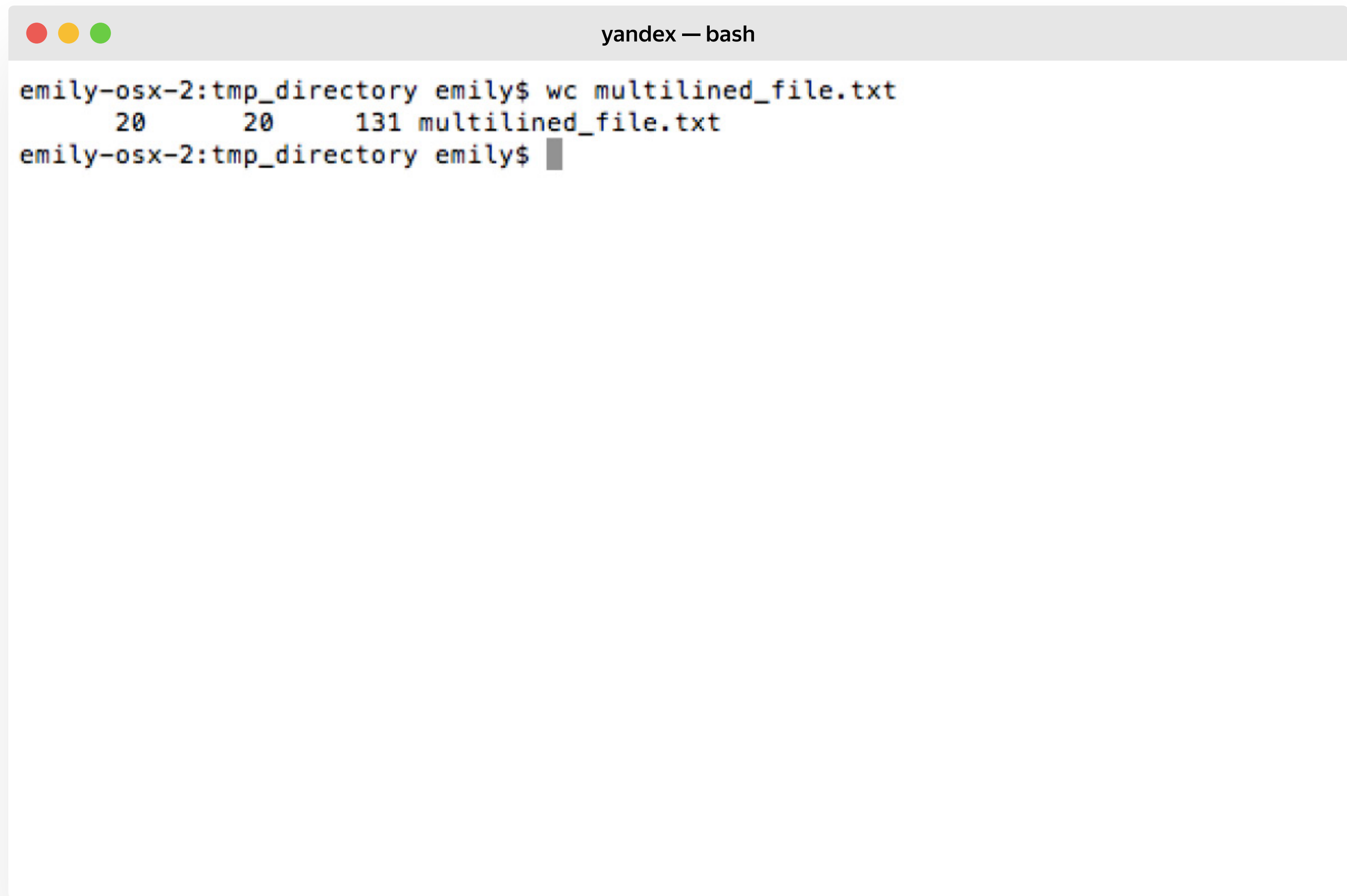
A terminal window titled "yandex — bash" displays the output of the less command. The output shows 20 lines of text, each labeled from "line1" to "line20". The last line of the output is "multilined_file.txt (END)", which is highlighted with a black background and white text. The terminal window has a standard macOS-style title bar with red, yellow, and green window control buttons.

WC

- › "word count" prints a count of lines, words, and bytes for each file

wc [OPTION]... [FILE]...

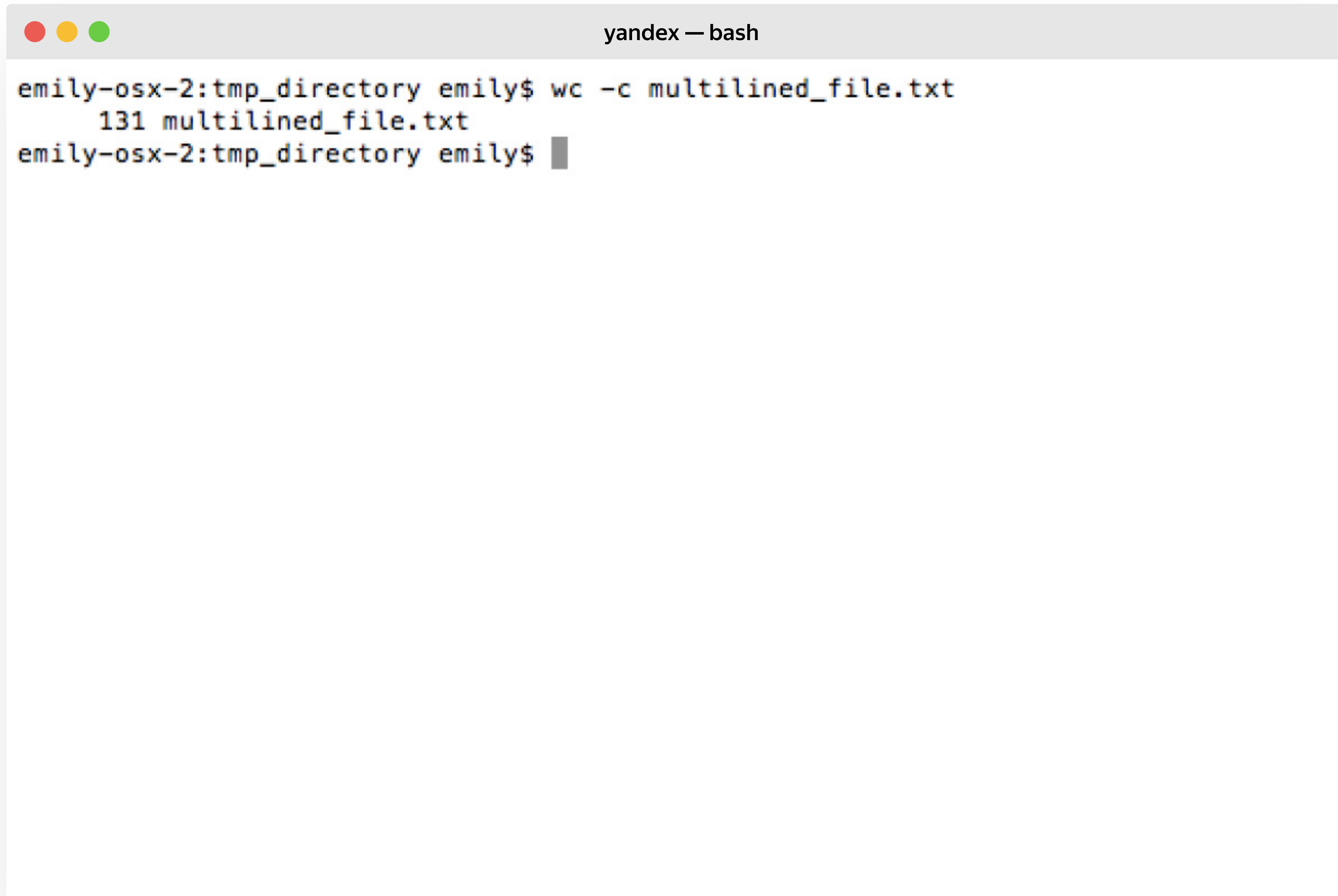
WC



A terminal window titled "yandex — bash" with standard macOS window controls (red, yellow, green buttons). The terminal shows the command `wc multilined_file.txt` being executed. The output is displayed on the next line, showing the number of lines, words, and bytes for the file. The prompt `emily-osx-2:tmp_directory emily$` is visible on both lines.

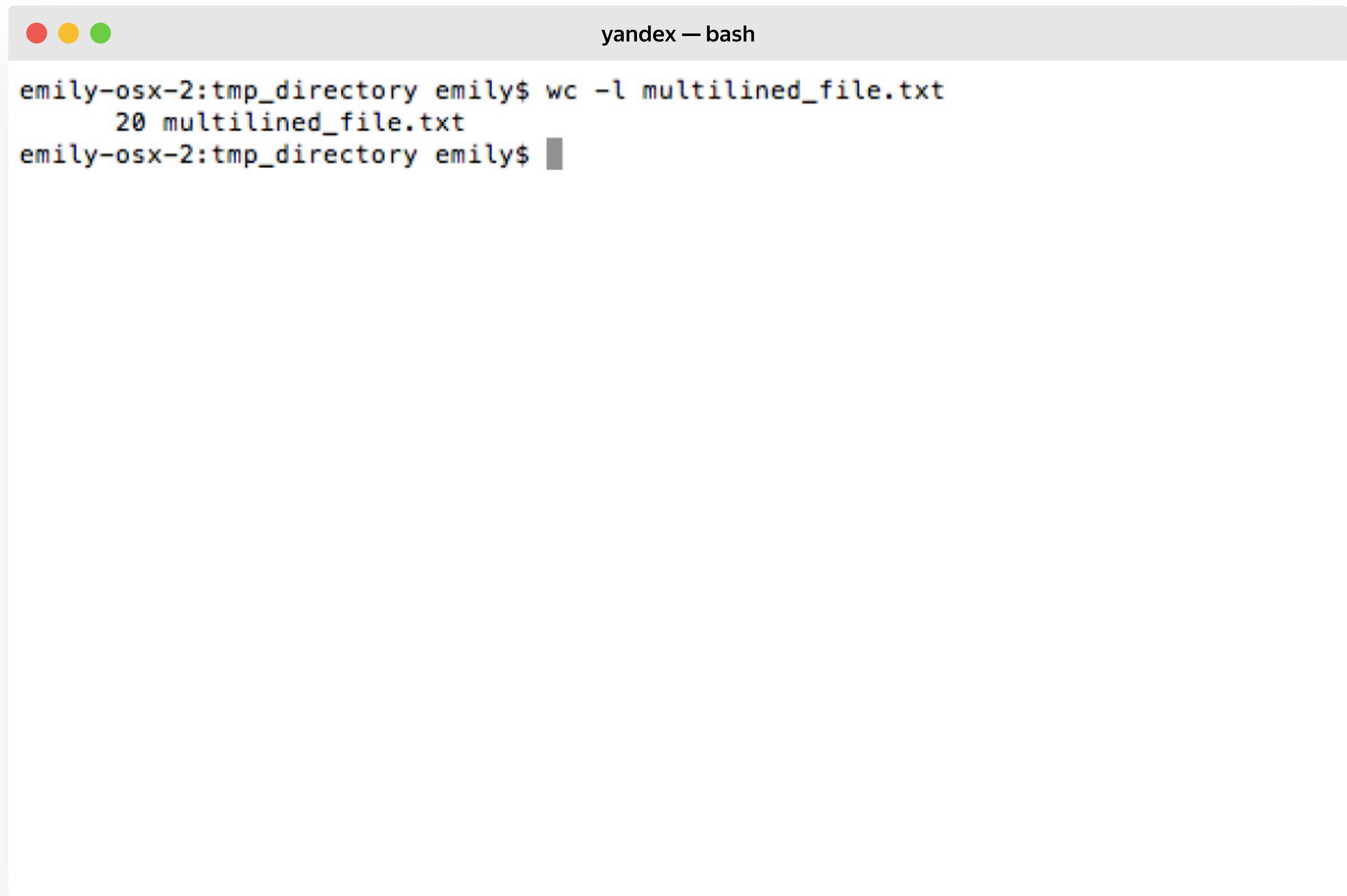
```
emily-osx-2:tmp_directory emily$ wc multilined_file.txt
    20     20    131 multilined_file.txt
emily-osx-2:tmp_directory emily$
```

WC -C

A terminal window titled "yandex — bash" with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$" followed by the command "wc -c multilined_file.txt". The output is "131 multilined_file.txt", and the prompt "emily-osx-2:tmp_directory emily\$" is shown again with a cursor.

```
emily-osx-2:tmp_directory emily$ wc -c multilined_file.txt
 131 multilined_file.txt
emily-osx-2:tmp_directory emily$
```

wc -l



```
yandex — bash  
emily-osx-2:tmp_directory emily$ wc -l multilined_file.txt  
  20 multilined_file.txt  
emily-osx-2:tmp_directory emily$
```

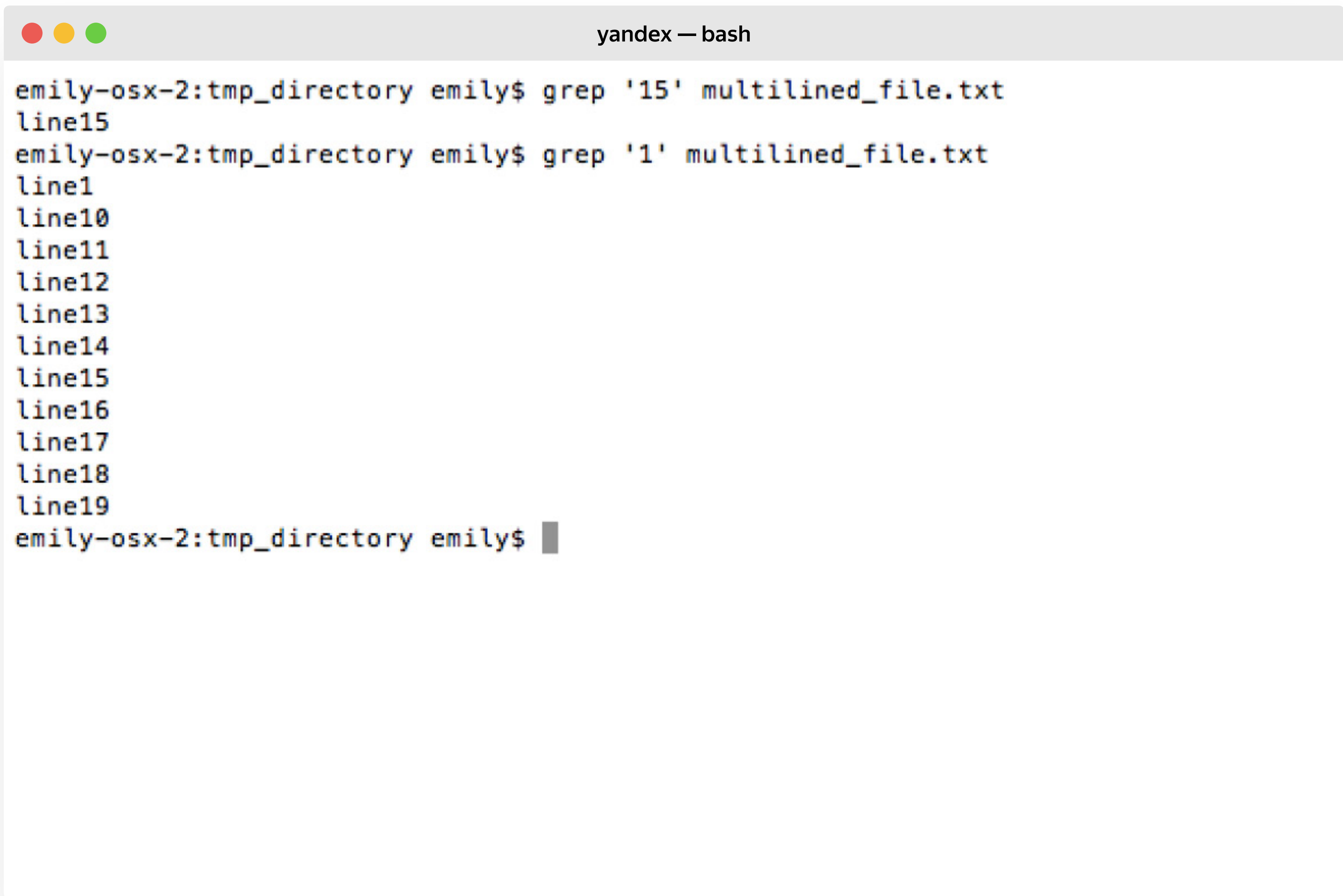
A terminal window titled "yandex — bash" with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal shows a command prompt "emily-osx-2:tmp_directory emily\$". The user enters the command "wc -l multilined_file.txt". The output is "20 multilined_file.txt". The prompt returns to "emily-osx-2:tmp_directory emily\$".

grep

- › "global regular expression print" processes text and prints any lines which match a specified pattern

grep [OPTIONS] PATTERN [FILE...]

grep



```
emily-osx-2:tmp_directory emily$ grep '15' multilined_file.txt
line15
emily-osx-2:tmp_directory emily$ grep '1' multilined_file.txt
line1
line10
line11
line12
line13
line14
line15
line16
line17
line18
line19
emily-osx-2:tmp_directory emily$
```

In-video quiz

- › How can we print first 17 lines of “temporary_logs.txt”?
- › How can we calculate number of lines for file “temporary_logs.txt”?
- › How can we print all lines from file “temporary_logs.txt”, that contains ‘data’ substring?

Outcome

- › For exploring files content we use the following commands:
 - › **cat** — print file content
 - › **head/tail** — print first/last part of files
 - › **more/less** — output file content by screens
 - › **wc** — “word count”
 - › **grep** — find any lines which match a specified pattern

BigDATAteam