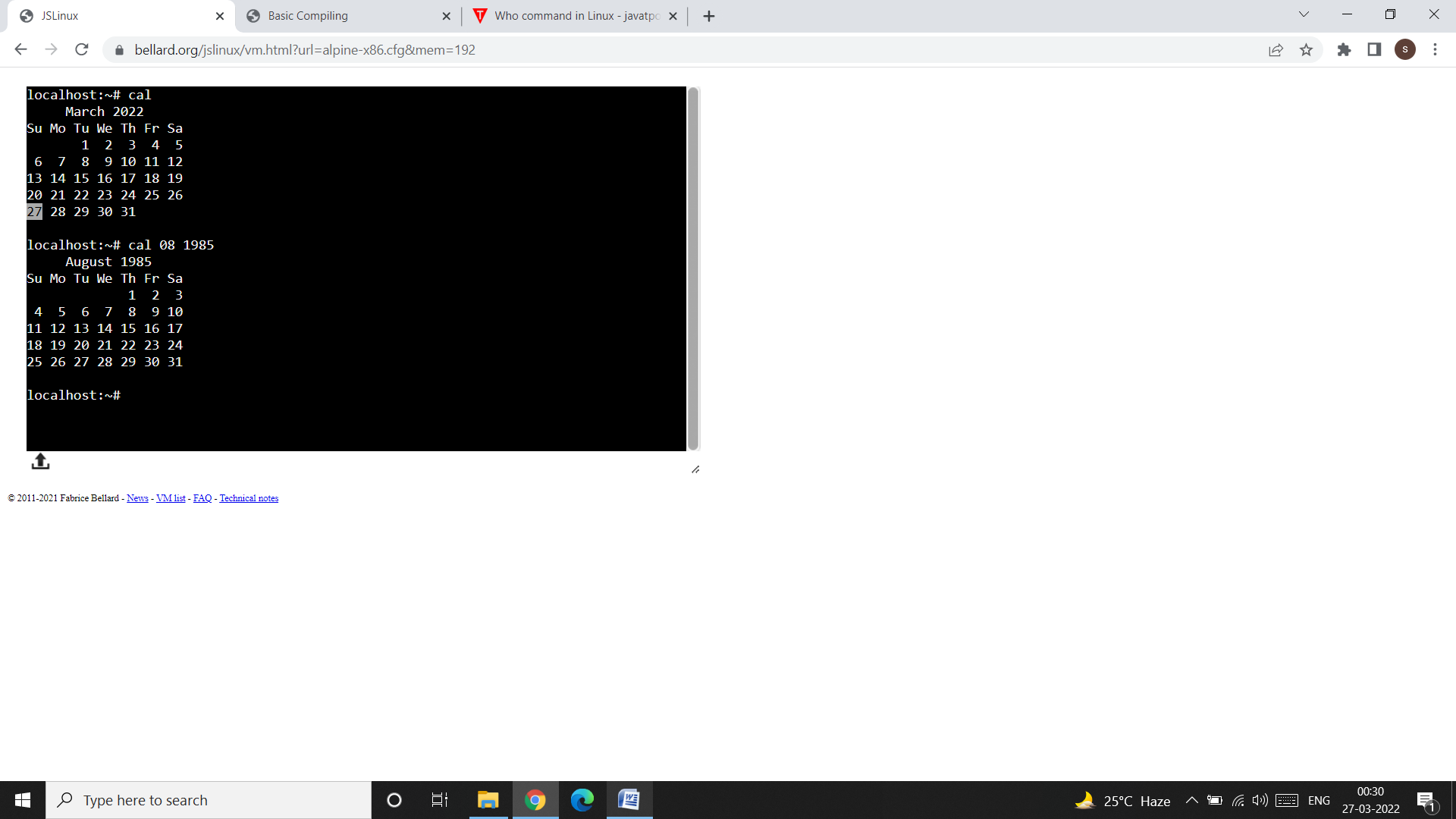
**How to run UNIX commands:**

* Install Unix/Linux operating system in your PC or go to the following link in your web browser: https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192.
* ↩ Enter Key Sign

**Unix General-Purpose Commands:**

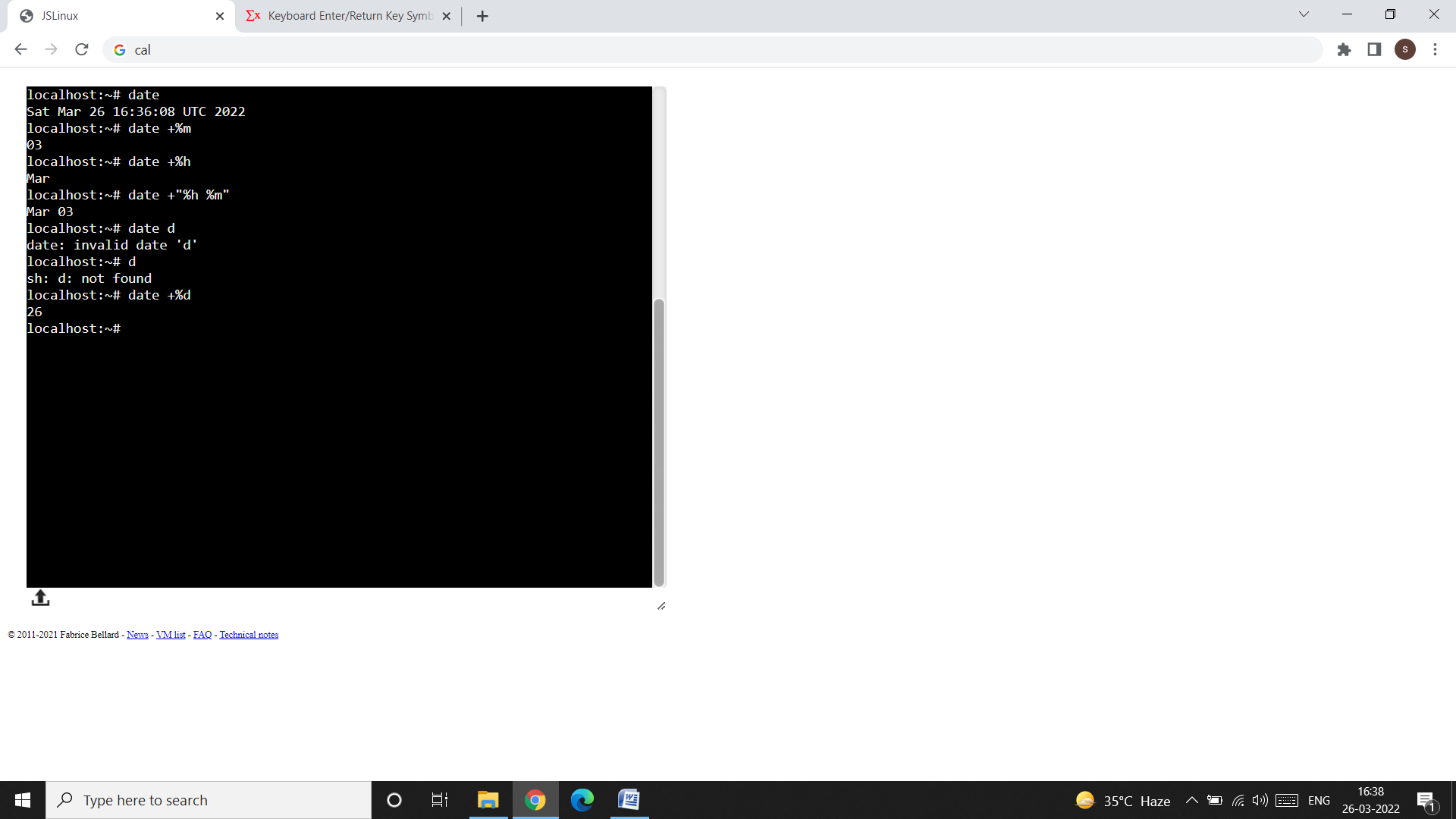
**1. cal: The Calendar:**

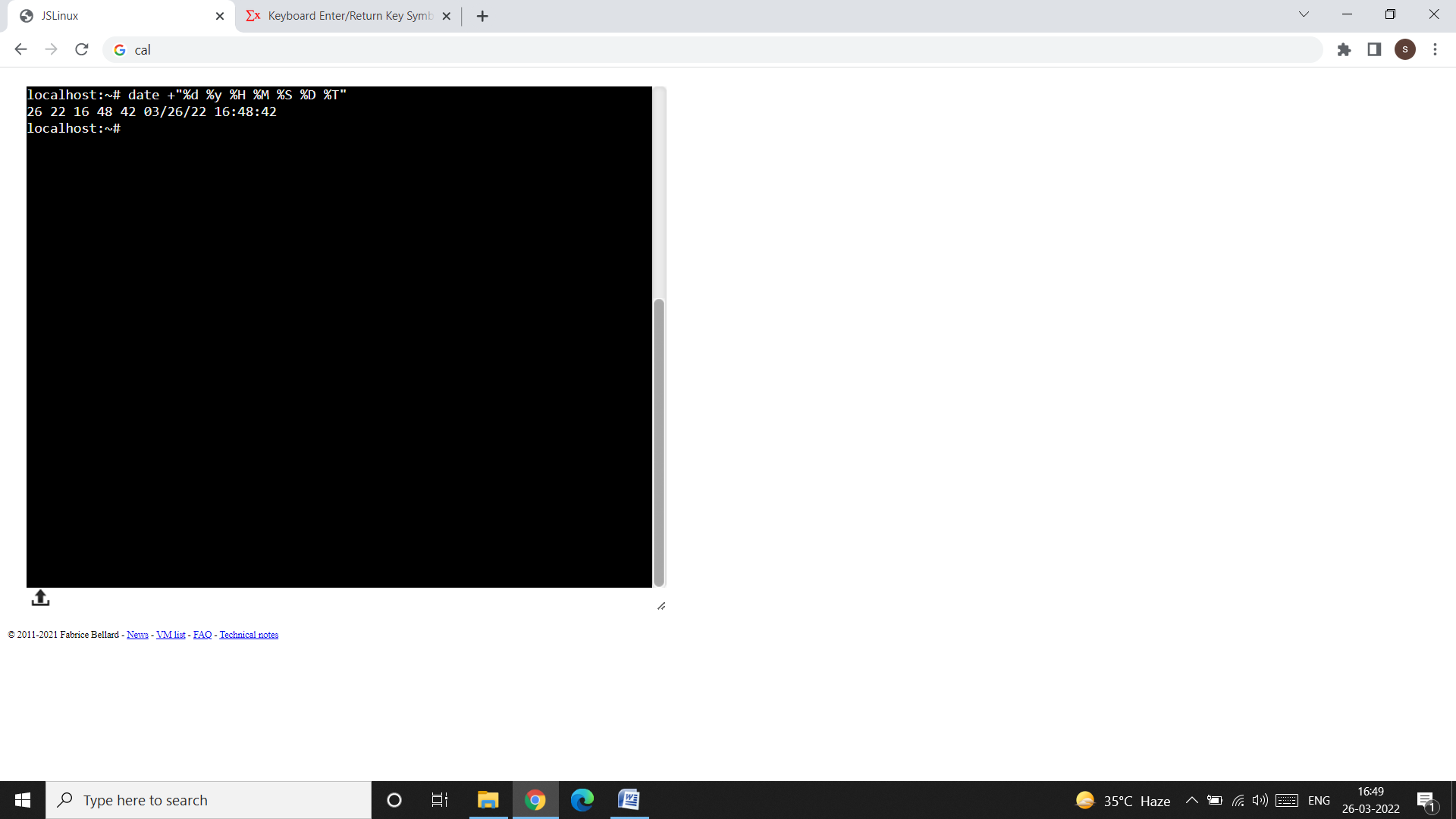
* You can invoke the cal command to see the calendar of any specific month or a complete year.
* Example: $ cal (↩ ) will give the calendar of current month and year (i.e., March 2022).
* $ cal 03 2006 will give the calendar of March 2006.



**2. date: Displaying the system date**

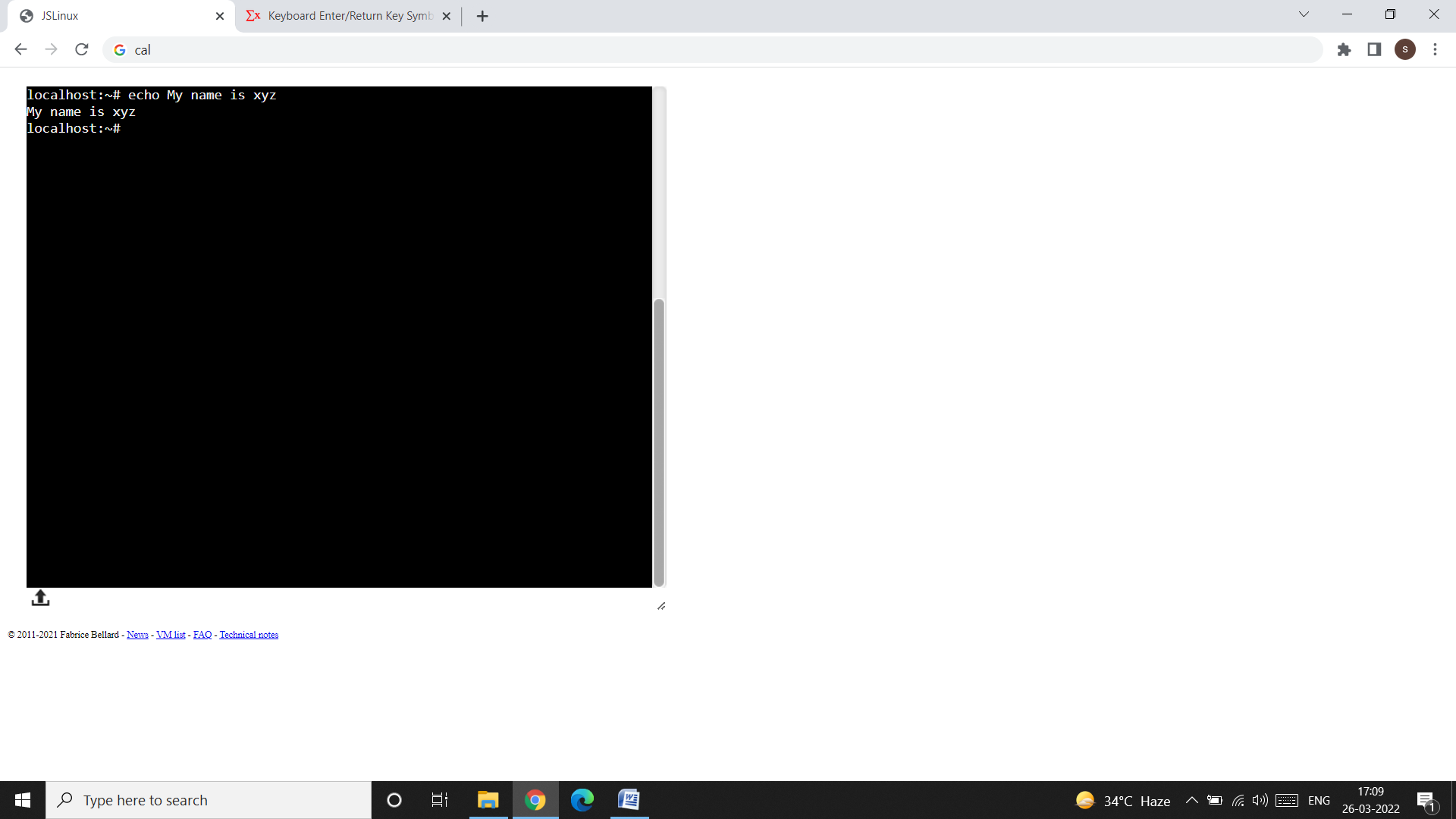
* You can display the current date with the date command, which shows the date and time to the nearest second.
* Example:$ date (↩ )



* There are many other format specifier, for example:
* d – The day of the month (1-31)
* y – The last two digits of the year.
* H, M, and S – The hour, minute, and second, respectively.
* D – The date in the format mm/dd/yy.
* T – The time in the format hh:mm:ss.
* When you use multiple format specifier, you must enclose them within quotes (single or double), and use a single + symbol before it. Example:
* 

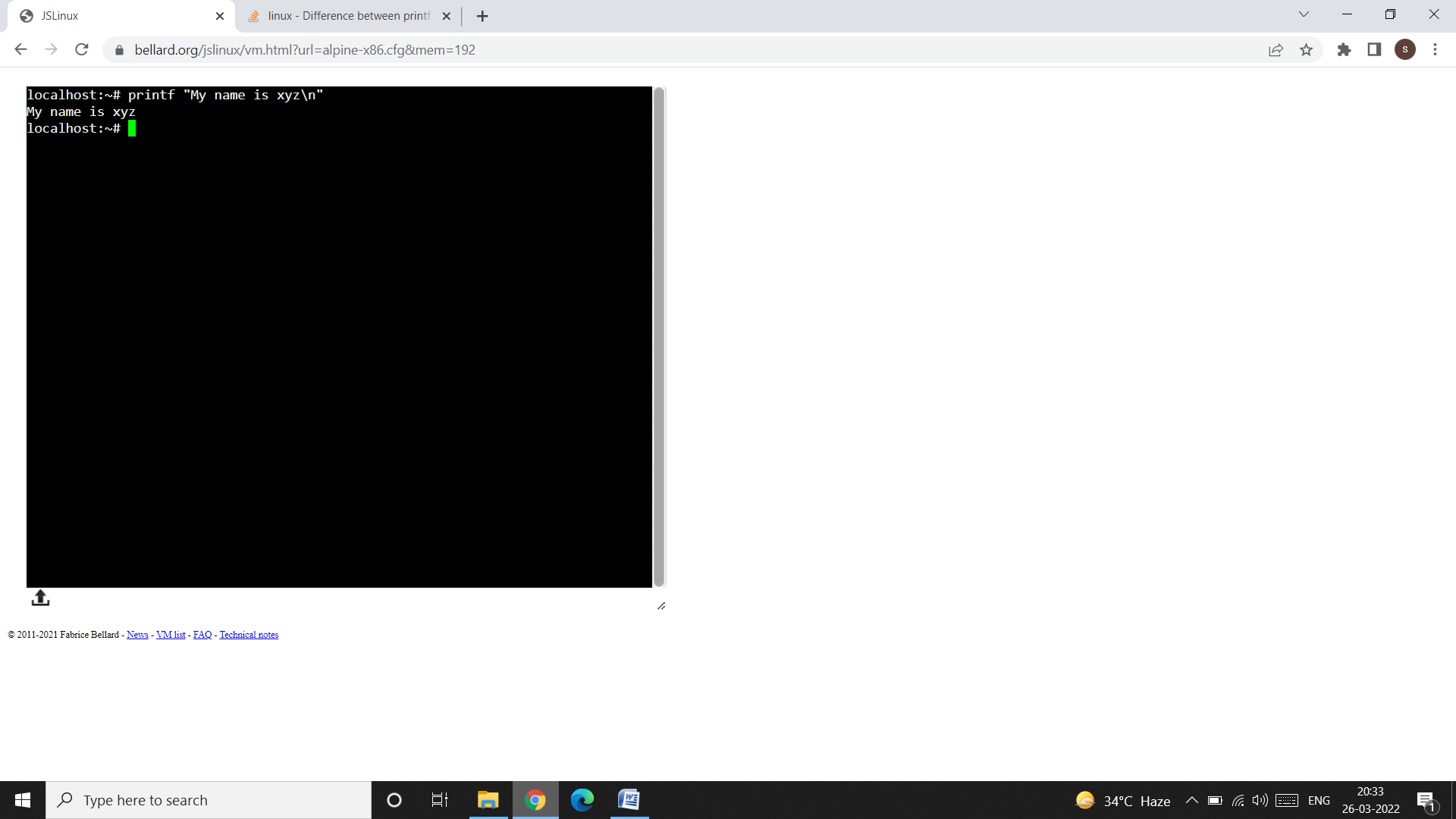
3. echo: Displaying a message

* This command is used to display messages on the terminal, or to issue prompts for taking user input.
* Example: $ echo My name is xyz

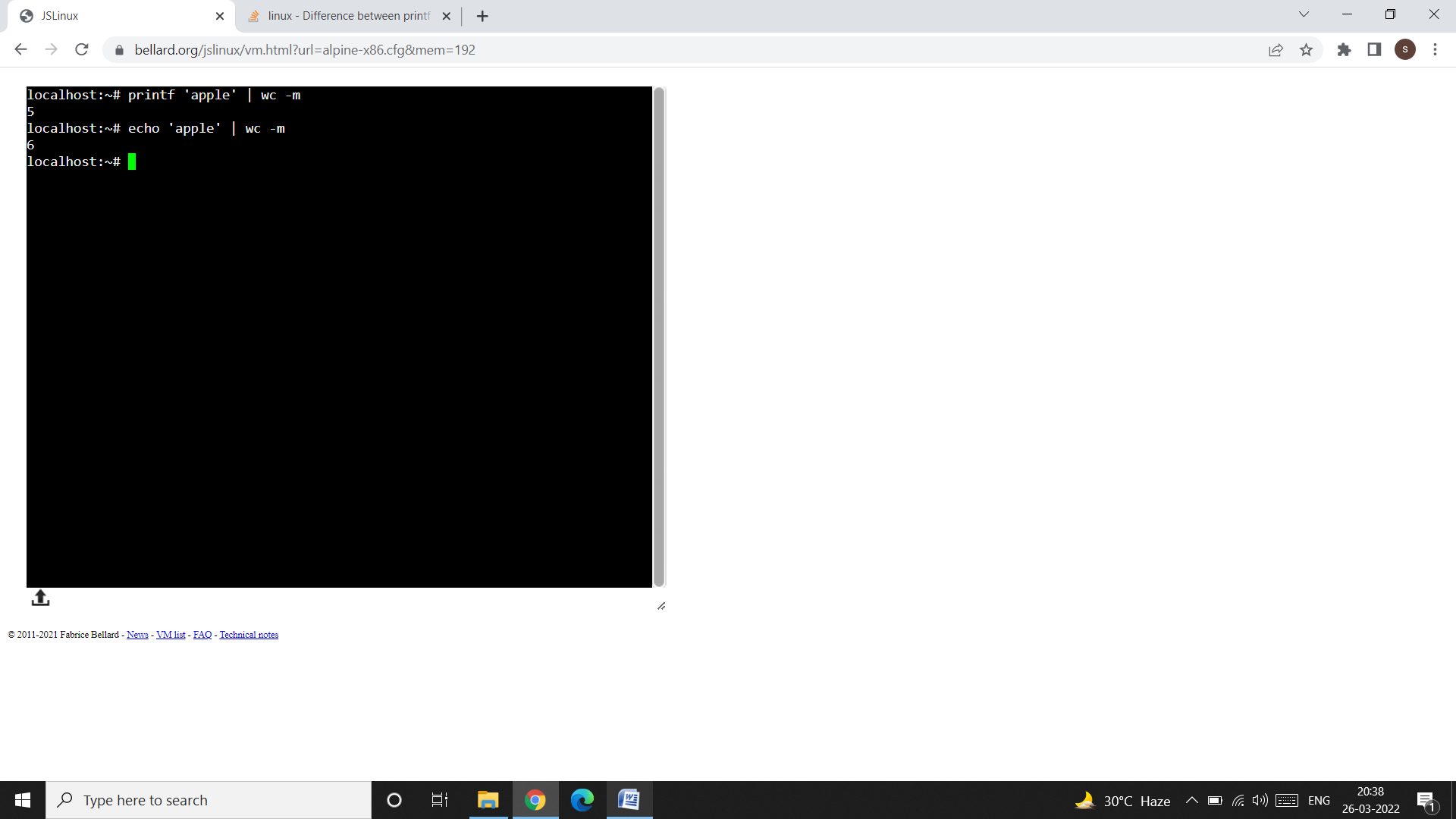


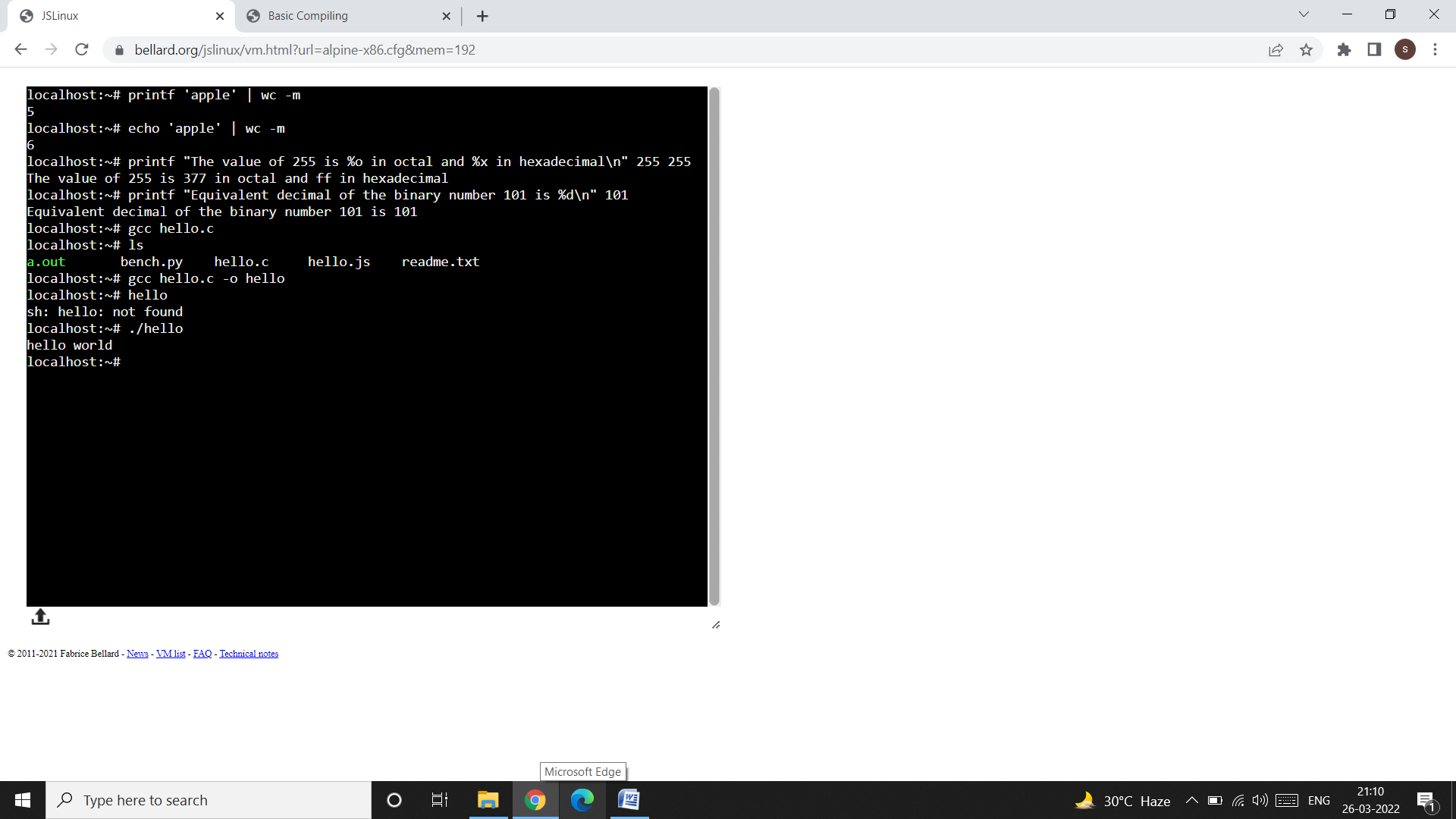
4. printf: An alternative to echo

* The printf command can be used in the same way as echo.



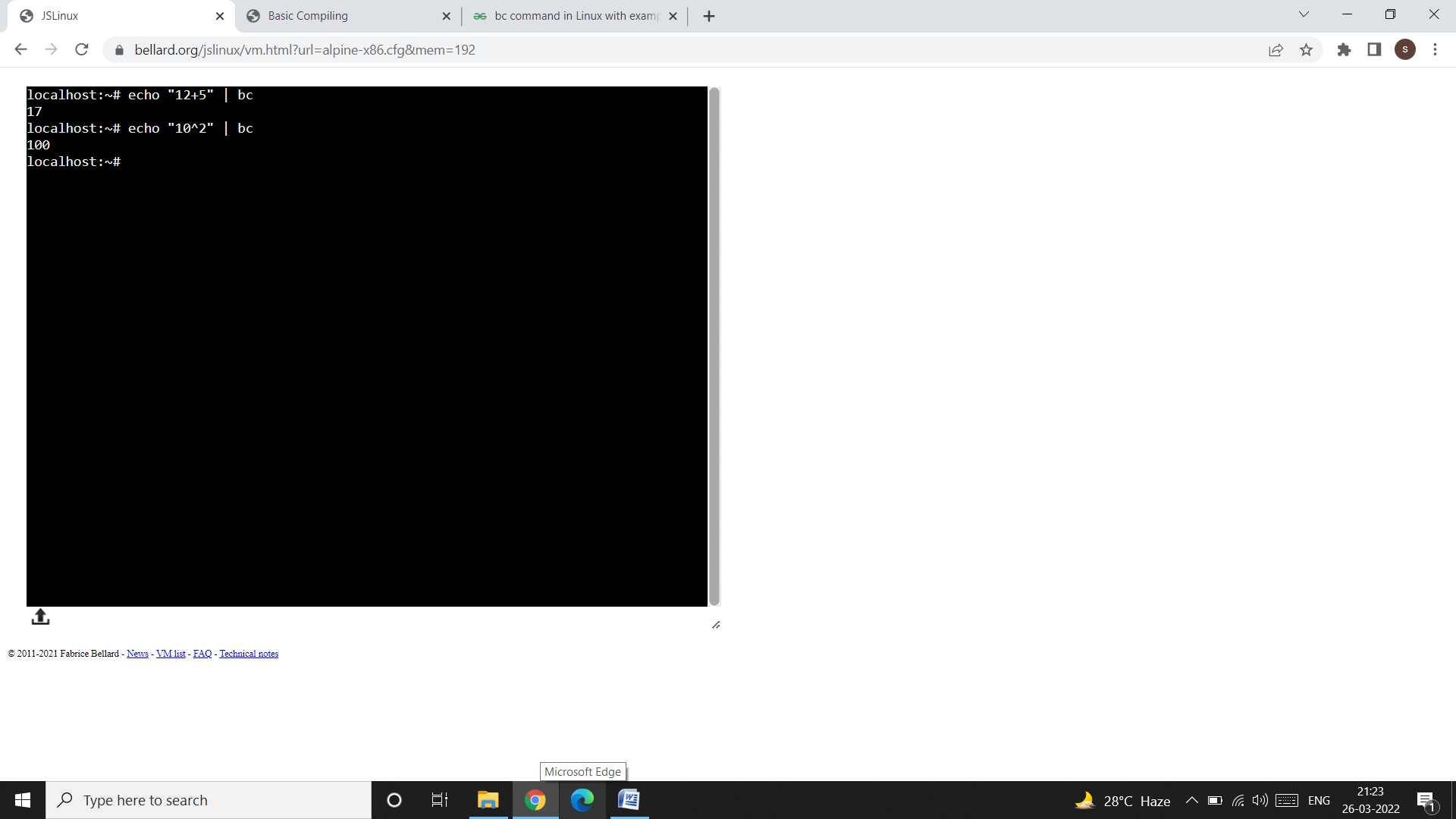
* **Difference between echo and printf:**
* To understand the difference between echo and printf consider the following commands:

****

* Why printf prints 5 and echo prints 6 characters?
* echo output the args, separated by spaces, **followed by a newline**, whereas printf only prints what appears in the format; it does not append an implied newline. That is why with echo we don’t have to use \n but with printf we have to use \n to move cursor to the new line.
* **Format strings with printf:**
* 
* %d – represents decimal integer
* %f – represents floating point number
* %s – represents string

5. bc: The calculator:

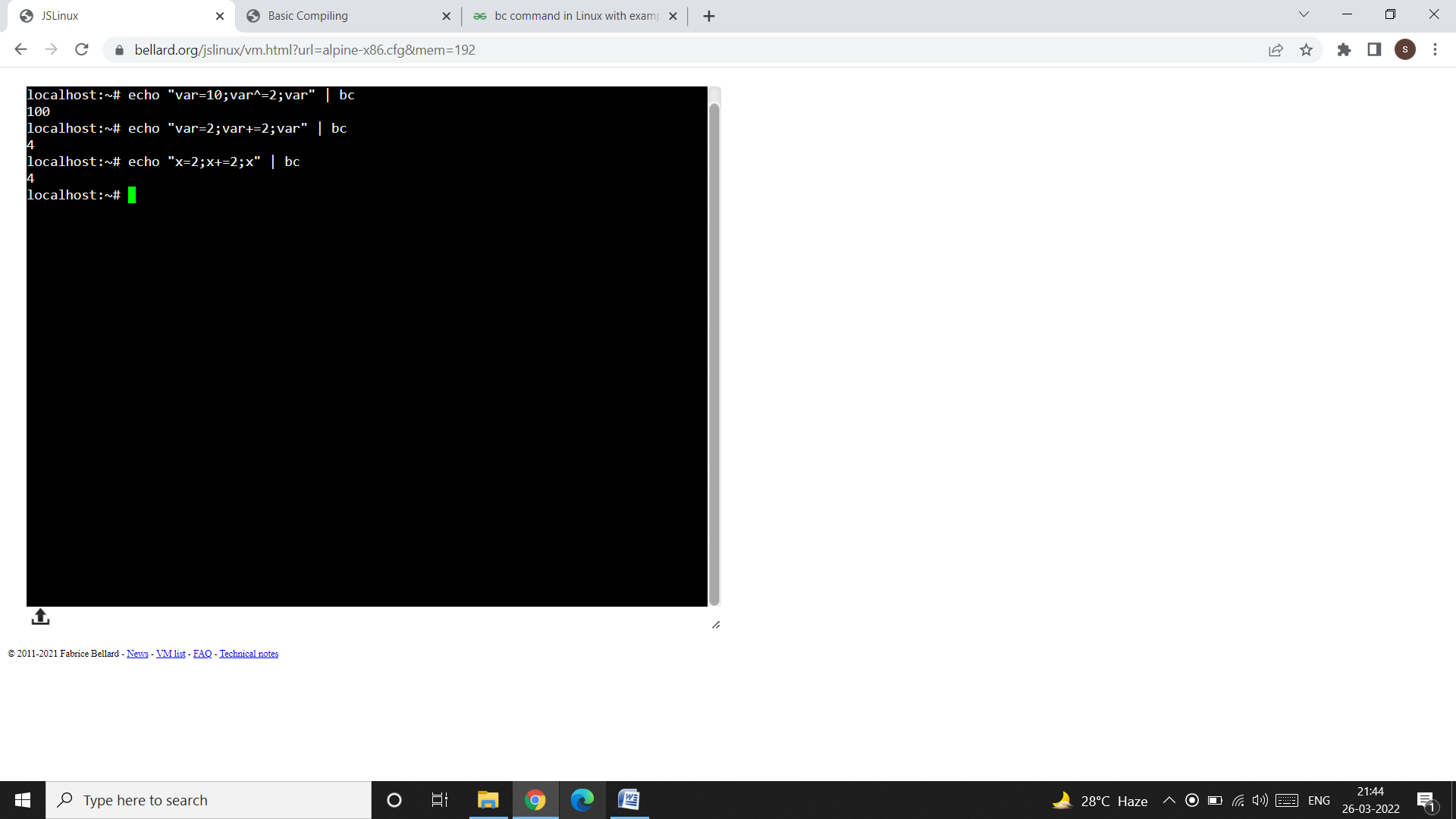
* The bc command is used for calculator operations. For example:



* **How to store the result of complete operation in variable?**

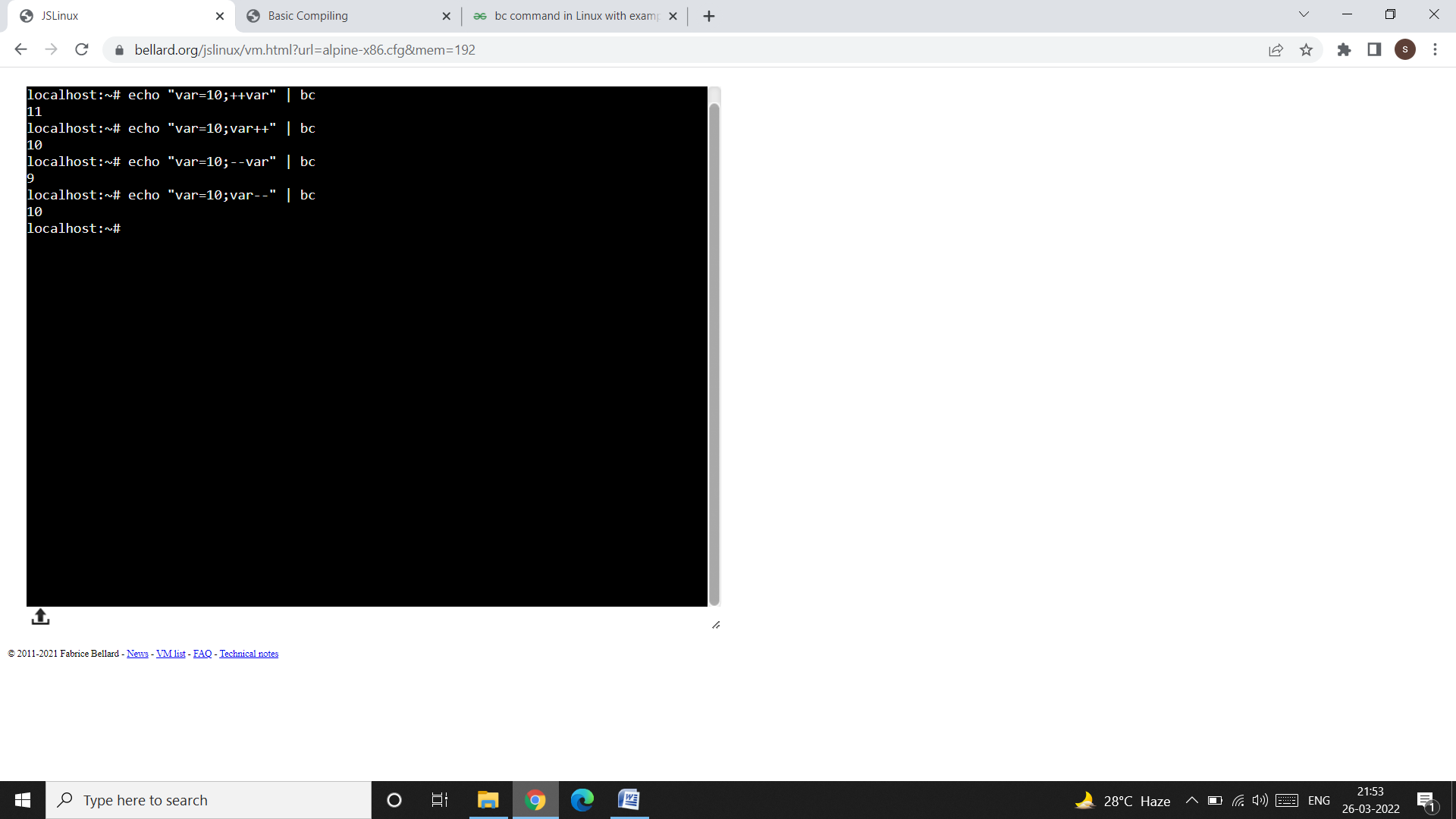
**Assignment Operators**   
The list of assignments operators supported are:

* **var = value** : Assign the value to the variable
* **var += value** : similar to var = var + value
* **var -= value** : similar to var = var – value
* **var \*= value** : similar to var = var \* value
* **var /= value** : similar to var = var / value
* **var ^= value** : similar to var = var ^ value
* **var %= value** : similar to var = var % value



**Increment Operators**   
There are 2 kinds of increment operators:

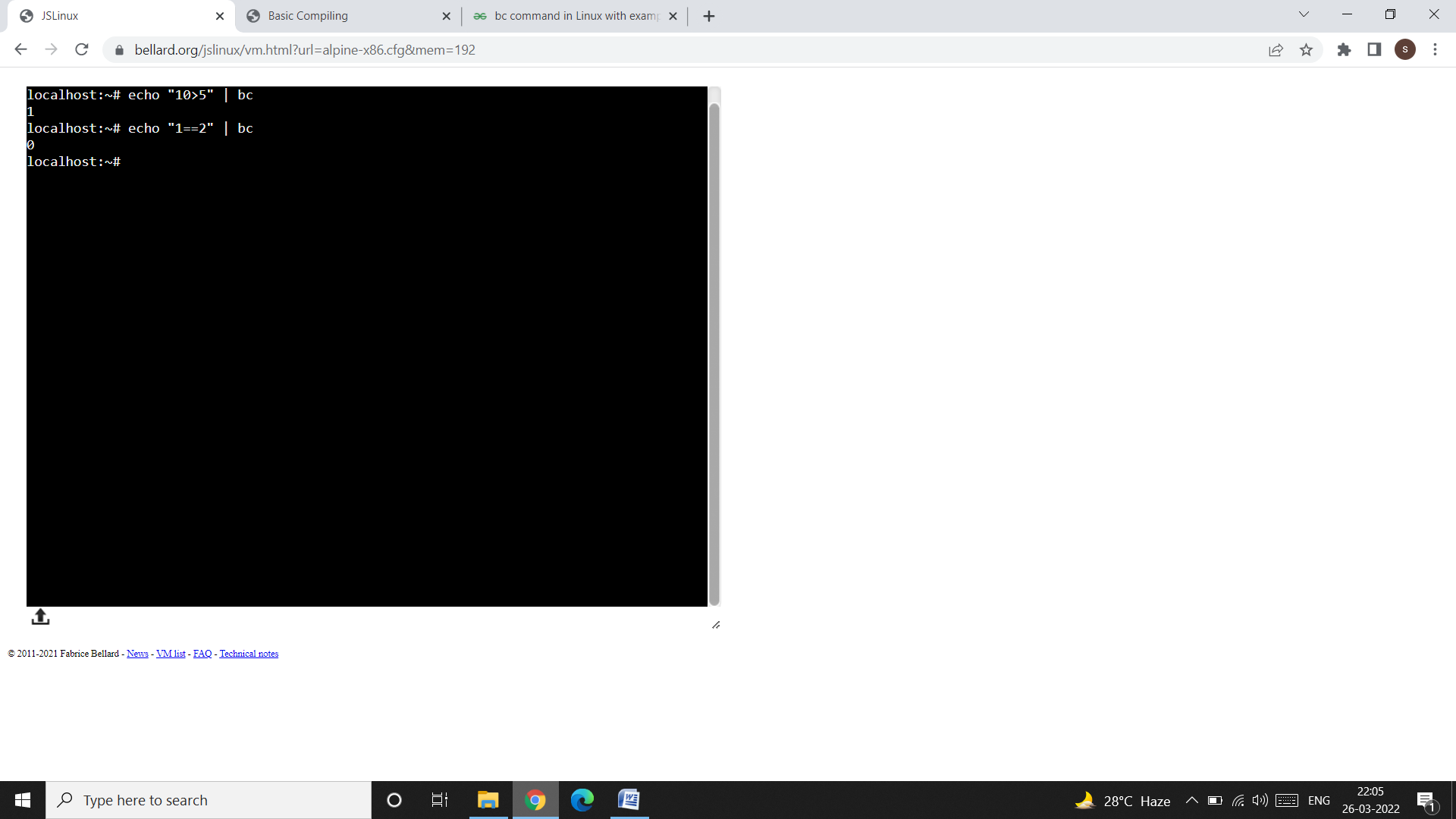
* **++var**: Pre increment operator, variable is increased first and then result of variable is stored.
* **var++**: Post increment operator, result of the variable is used first and then variable is incremented.
* **Decrement Operators**
* There are 2 kinds of decrement operators:
* **– – var**: Pre decrement operator, variable is decreased first and then result of variable is stored.
* **var – –** : Post decrement operator, result of the variable is used first and then variable is decremented.



**Comparison or Relational Operators**   
Relational operators are used to compare 2 numbers. If the comparison is true, then result is **1**. Otherwise(false), returns **0**. These operators are generally used in conditional statements like **if**.

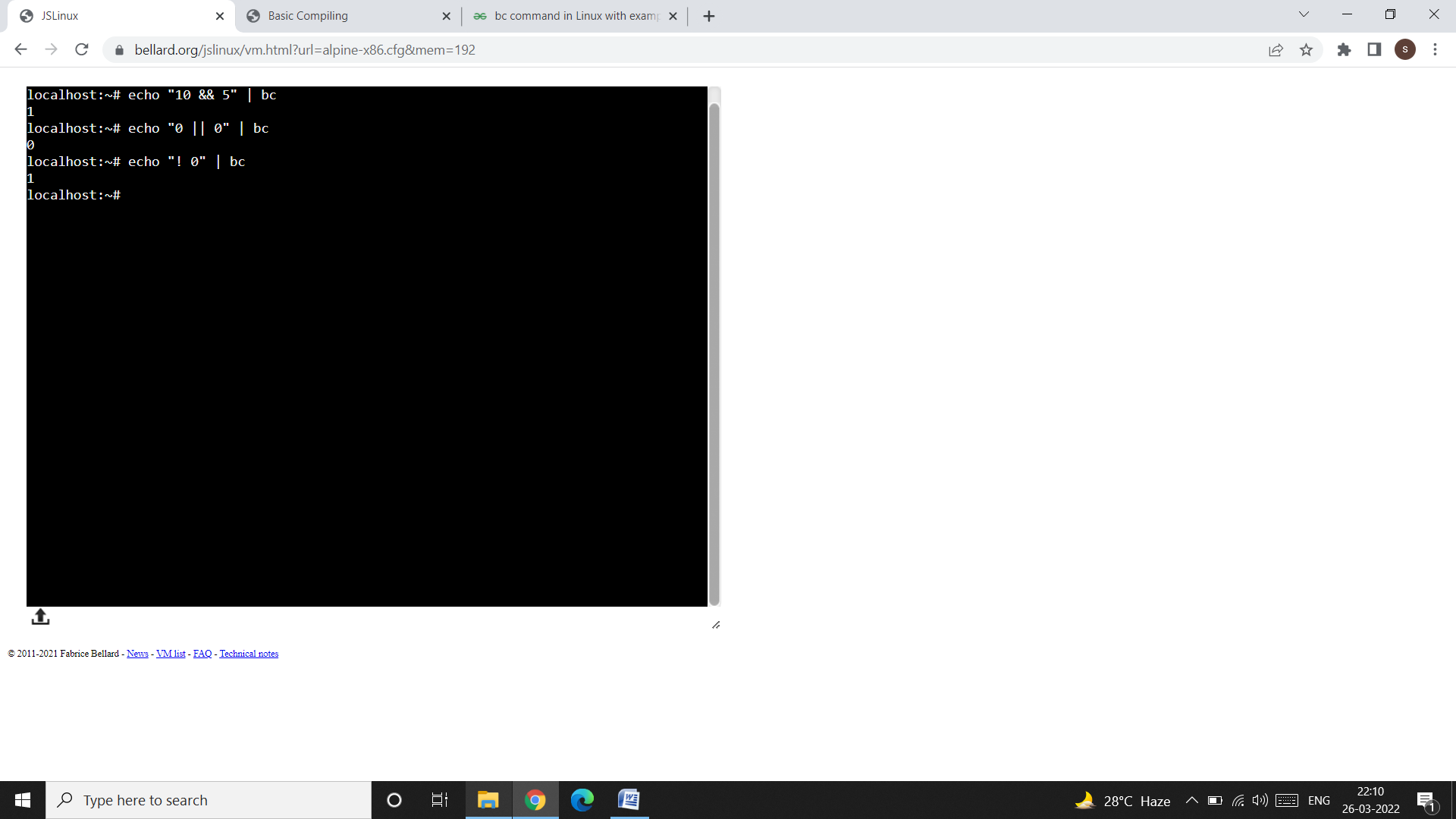
The list of relational operators supported in bc command are shown below:

* **expr1<expr2** : Result is 1 if expr1 is strictly less than expr2.
* **expr1<=expr2** : Result is 1 if expr1 is less than or equal to expr2.
* **expr1>expr2** : Result is 1 if expr1 is strictly greater than expr2.
* **expr1>=expr2** : Result is 1 if expr1 is greater than or equal to expr2.
* **expr1==expr2** : Result is 1 if expr1 is equal to expr2.
* **expr1!=expr2** : Result is 1 if expr1 is not equal to expr2.



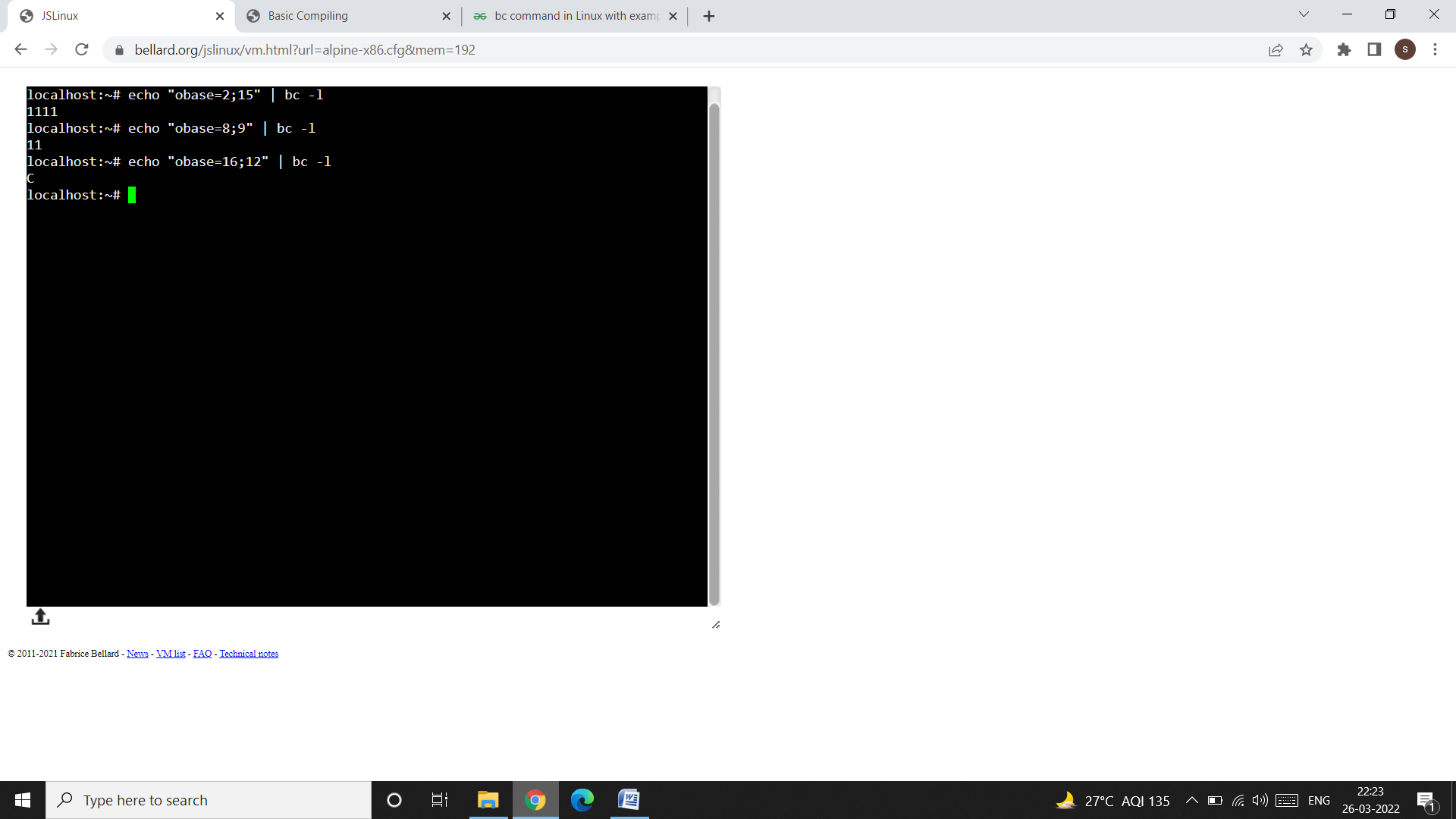
**Logical or Boolean Operators**   
Logical operators are mostly used in conditional statements. The result of the logical operators is either **1**(TRUE) or **0**(FALSE).

* **expr1 && expr2** : Result is 1 if both expressions are non-zero.
* **expr1 || expr2** : Result is 1 if either expression is non-zero.
* **! expr** : Result is 1 if expr is 0.

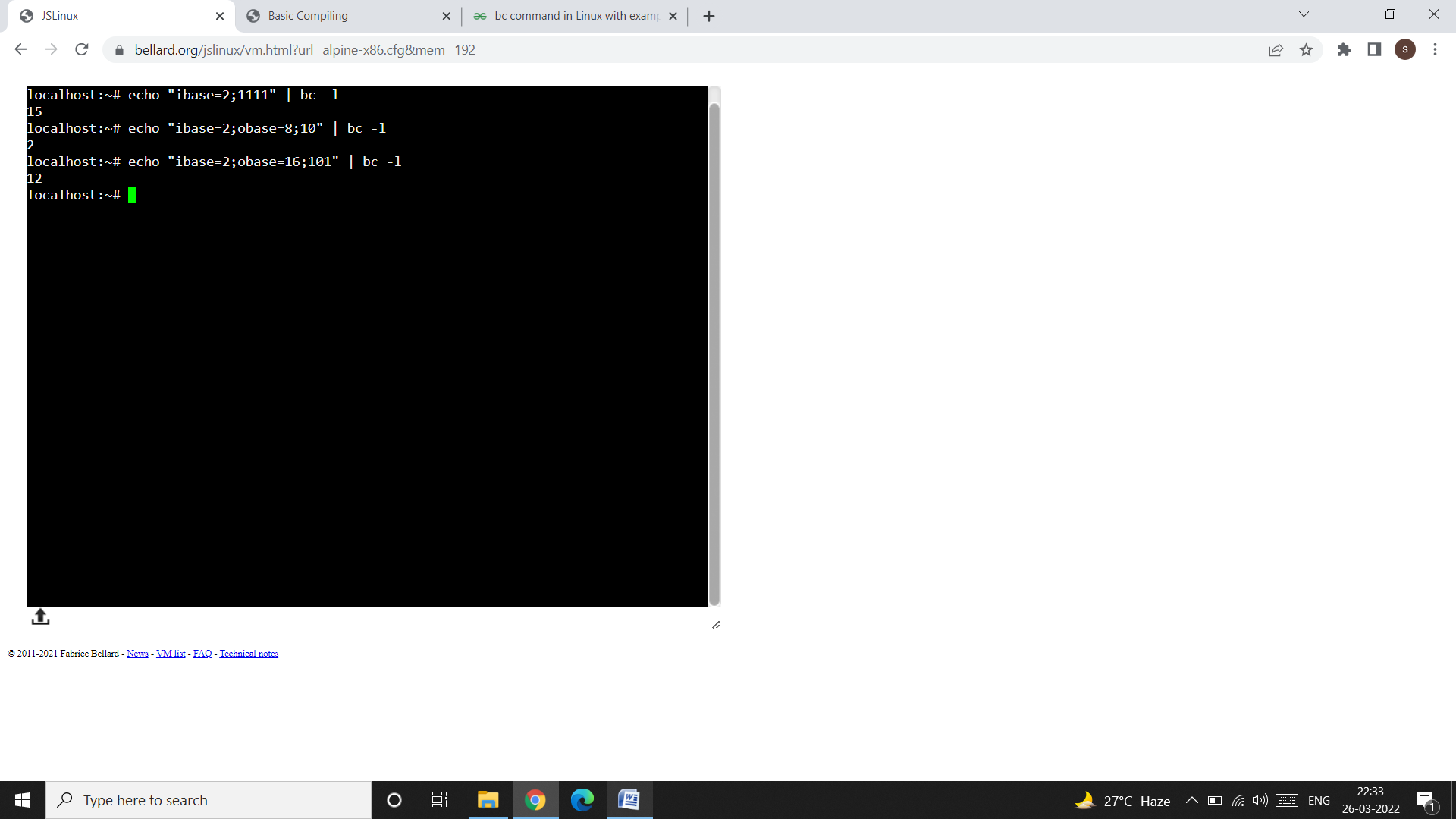


Conversion from one number system to other:

* The default base is **decimal.**
* ibase = input base, obase = output base.
* **Decimal to binary, octal, and hexadecimal conversions:**

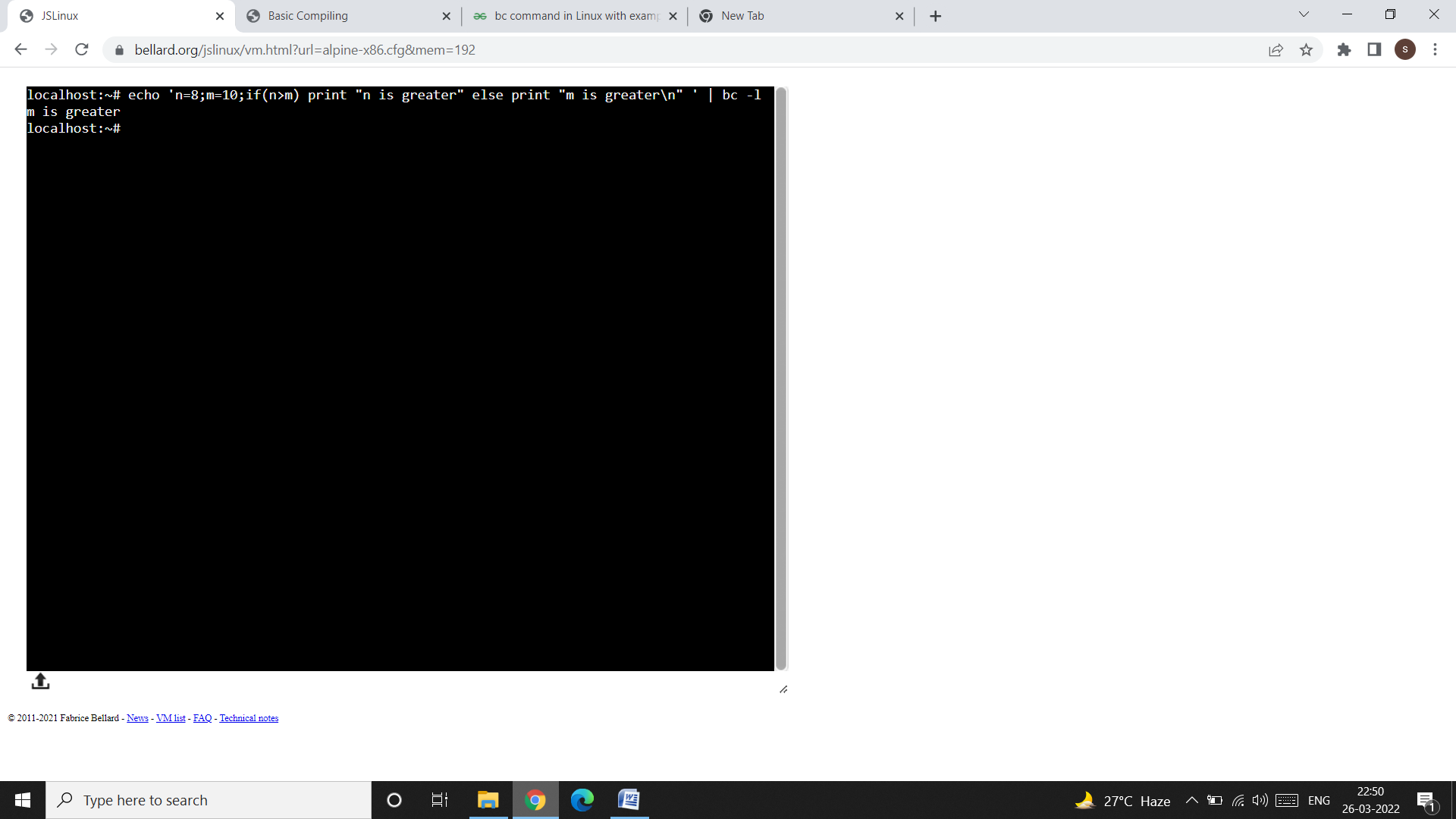
****

* **Convert Binary to Decimal, Octal, and Hexadecimal:**

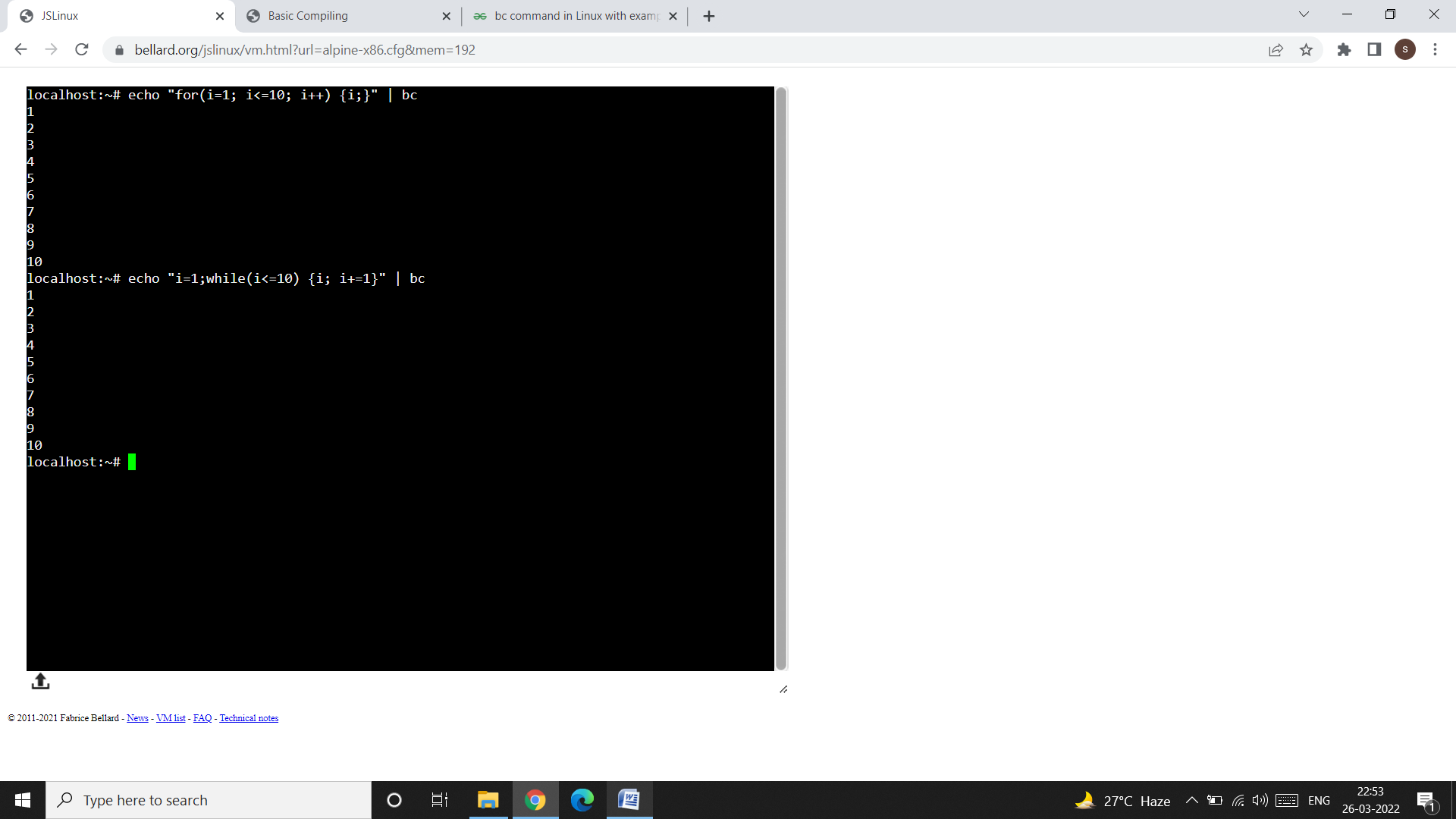
****

**Conditional Statements**

* Conditional Statements are used to take decisions and execute statements based on these decisions. bc command supports the if condition.

****

* **Iterative statements**bc command supports the for loop and while loop for doing iterations.



**We can write our arithmetic expressions in a file and then execute those statements by providing the filename to the bc command.**

Input :

$ cat >> example.txt

2+5;

var = 10\*3

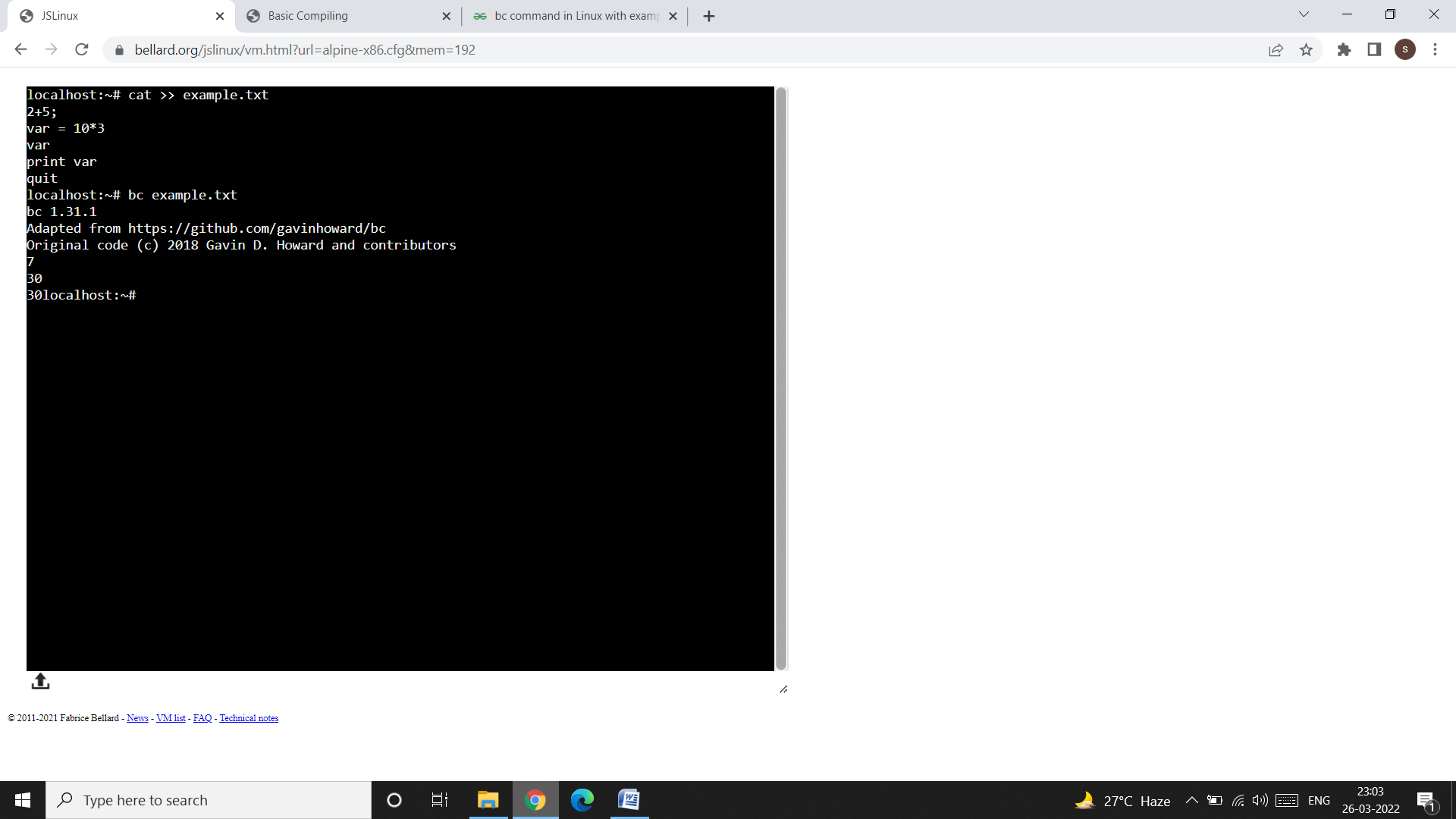
var

print var

quit

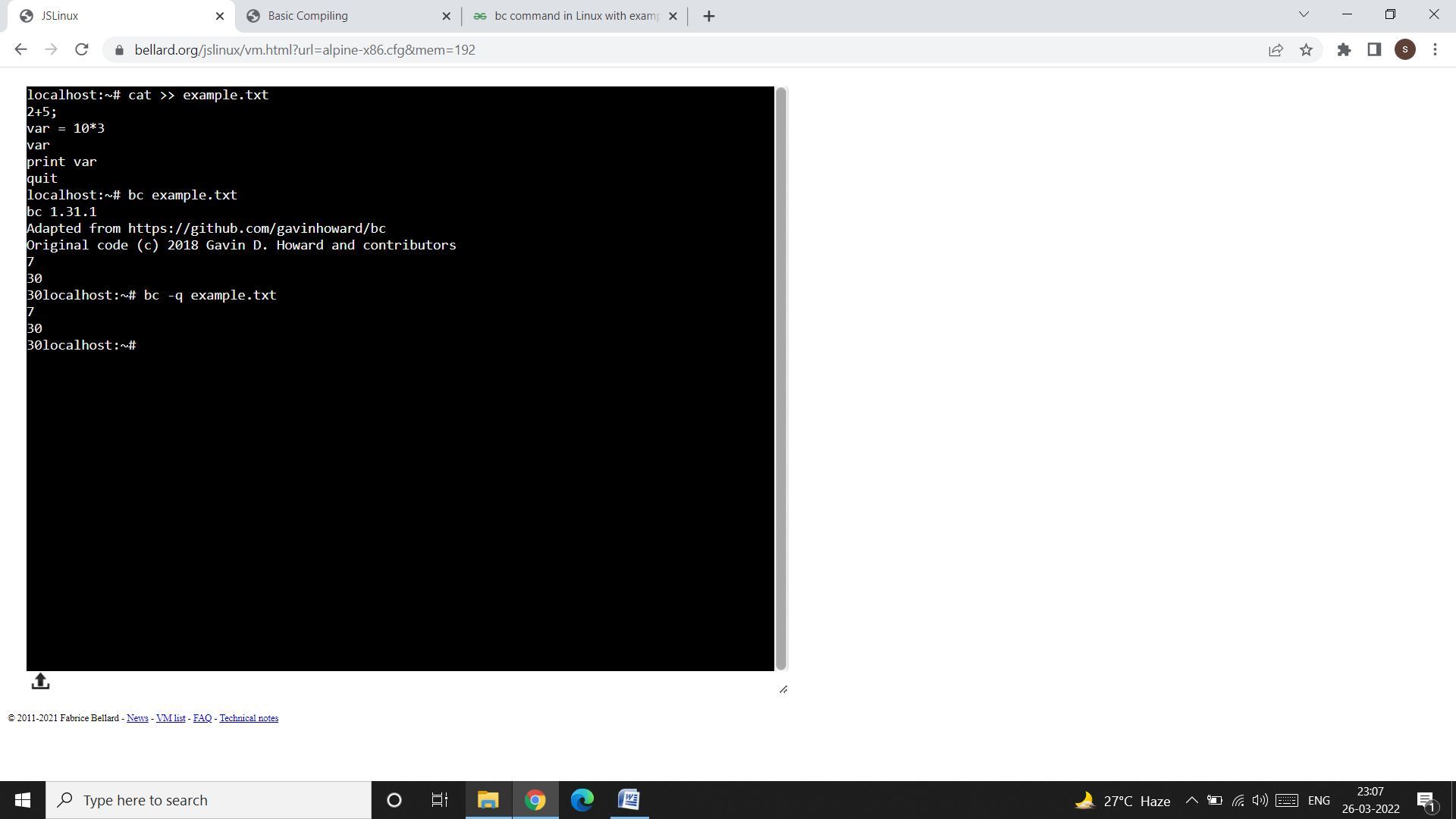
Press ctrl+D

$ bc example.txt



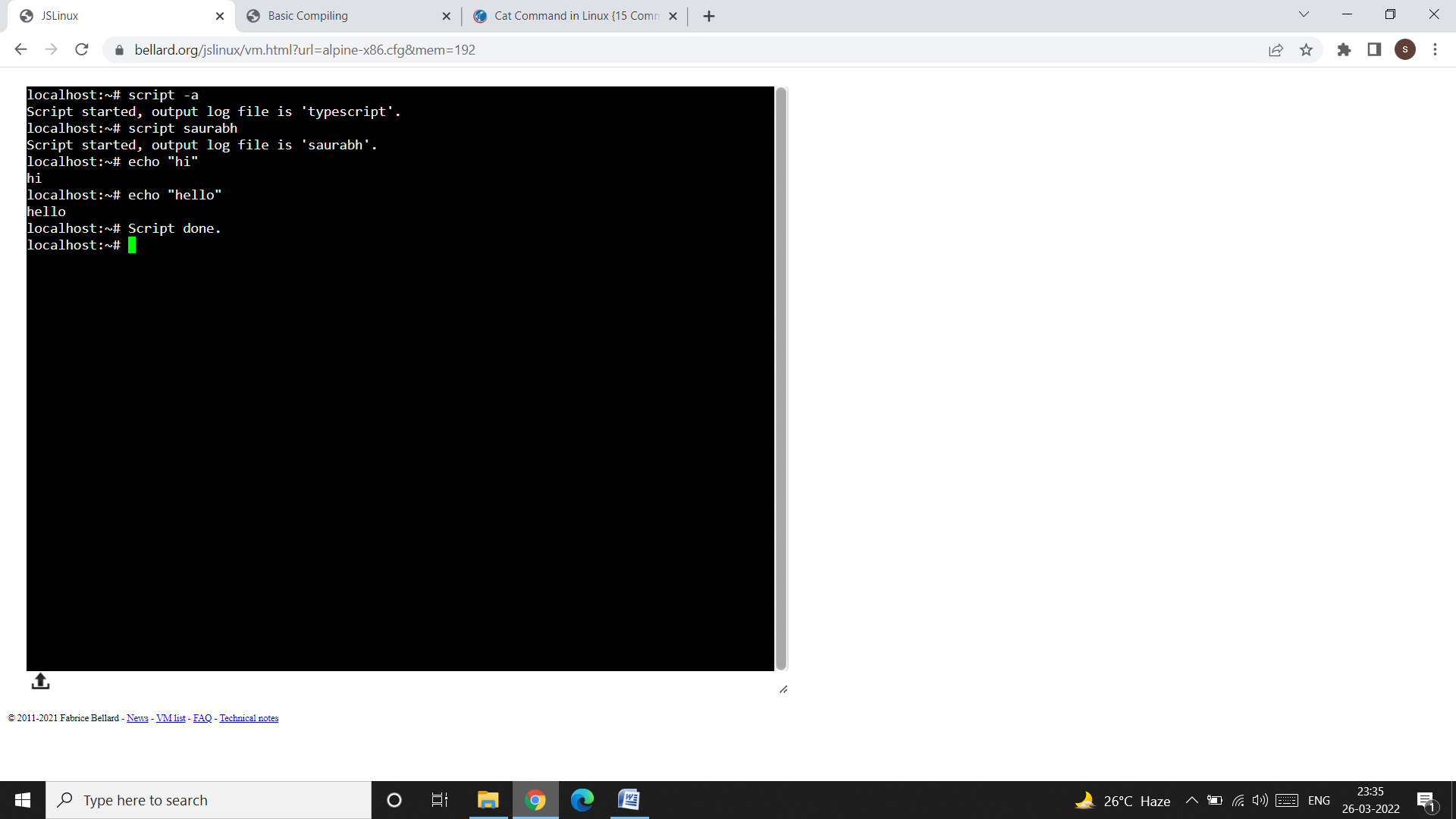
**TO AVOID SYSTEM GENERATED MESSAGE ON OUTPUT SCREEN, USE:**

$ bc -q example.txt

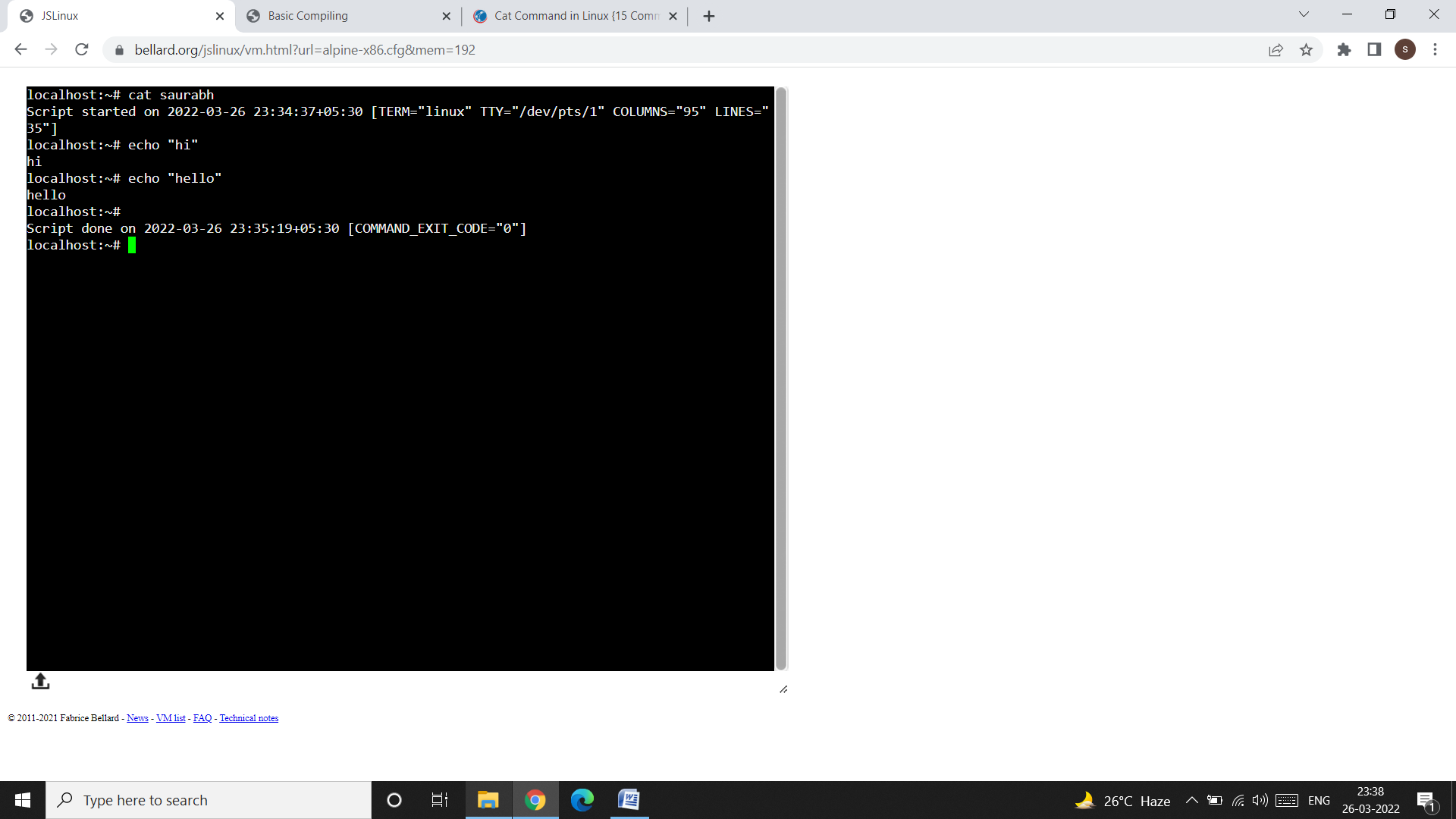


**6. script: Recording your session:**

* **script** command is useful to store in a file all keystrokes as well as output and error messages.
* We can later view the file using the **cat** command.
* If you are doing some important work and wish to keep a log of all your activities, you should invoke this command.

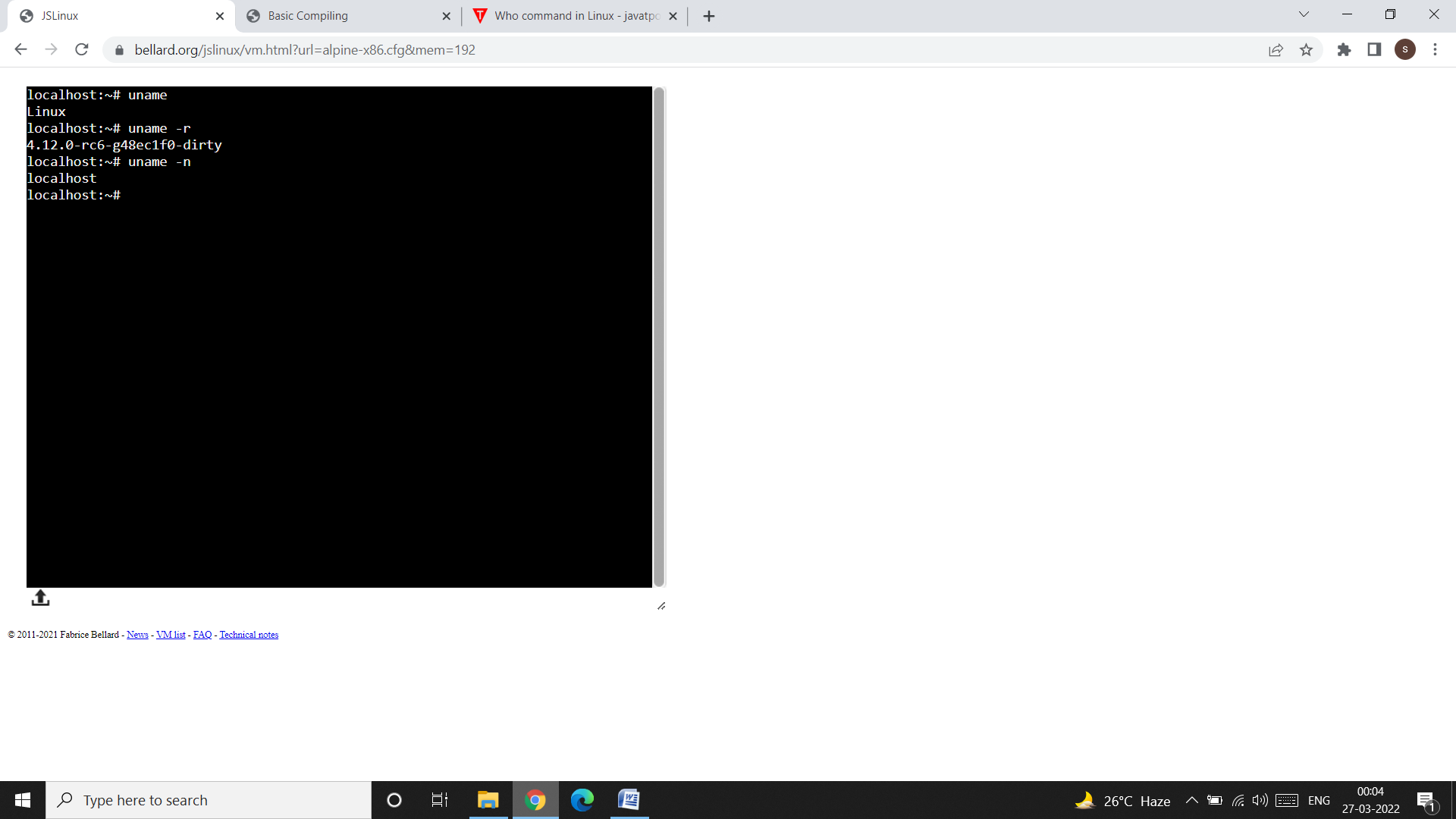


* You can view your entire recorded session or the content of your file using cat command like this:

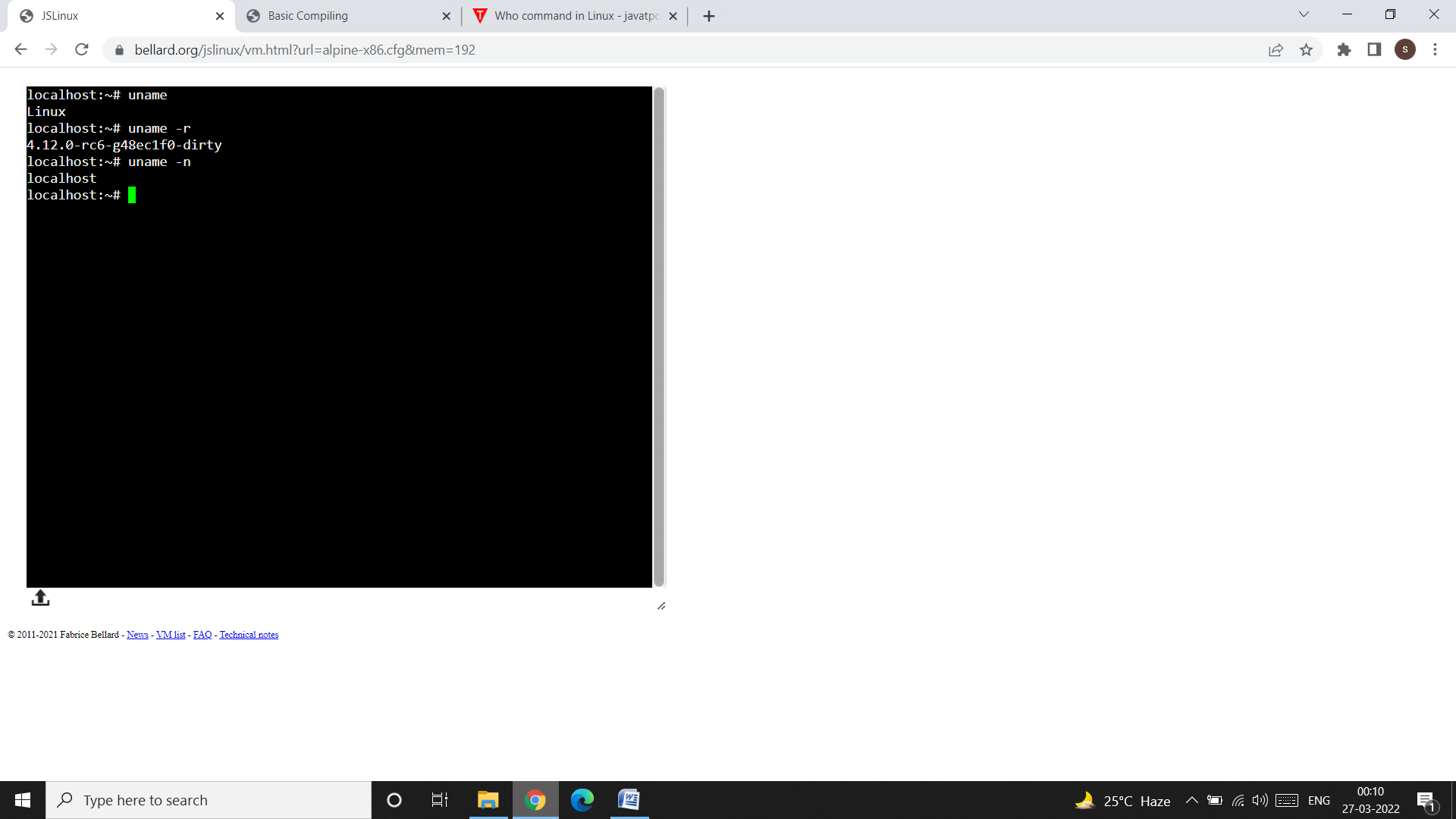


**7. uname: knowing your machine’s characteristics:**

* The uname command displays certain features of the operating system running on your machine. By default, it simply displays the name of the OS.

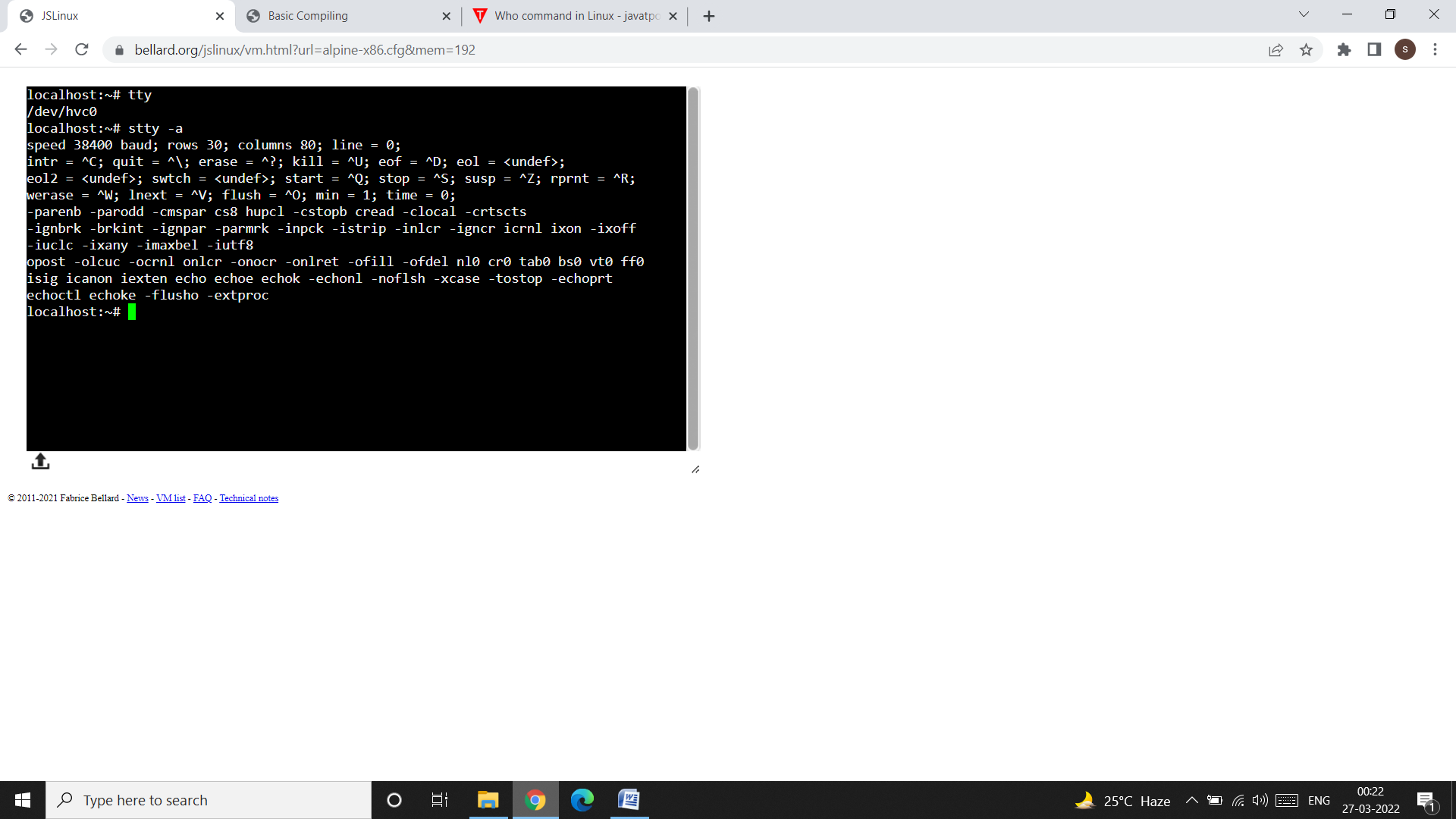


* To find the **version** and **host-name** of your operating system use uname –r and uname –n command respectively.



**8. tty: knowing your terminal:**

* tty (full-form: **teletype**) command tells you the **filename** of the **terminal** you are using.
* The **stty** command is used to display and set terminal characteristics.



* The output of the tty command is /dev/hvc0, where hvc0 is the terminal’s file name and dev is the name of the directory in which your terminal is residing.
* The –a (all) option in the stty command displays the current setting of your terminal.