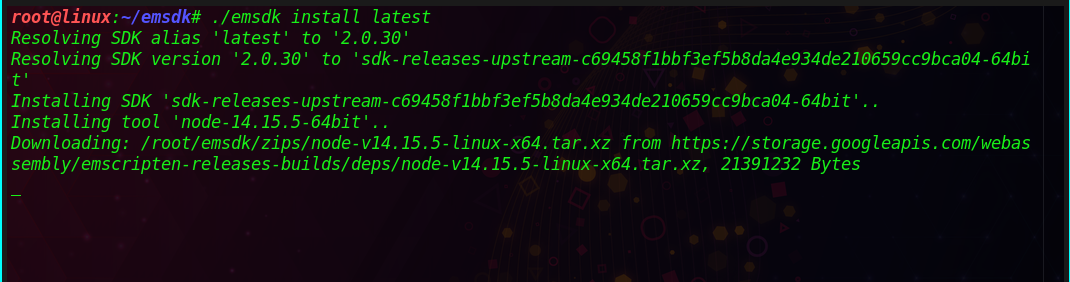
Development Environment

Core Emscripten is a python script whose repository can be cloned from GitHub sources.



Figure 1:GitHub clone.

The latest SDK tools can be install using ‘install latest’ command in the cloned repository.

Figure 2:Install SDK tools.

Latest installed SDK can be made available to current user by activating. Instructions are written to ‘.emscripten’ file located at the current user’s directory.

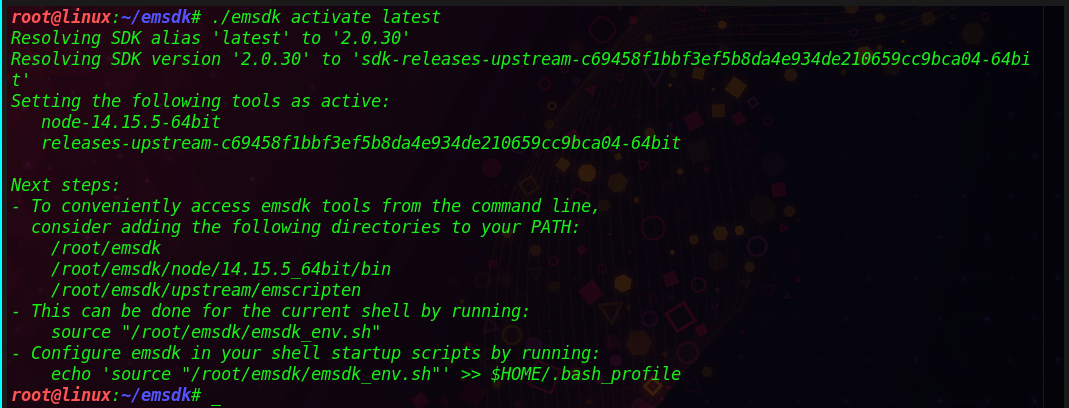


Figure 3:Activating latest SDK.

Active PATH and environmental variables for the current terminal available in the user context are set using source command.

Figure 4:Environmental variables.

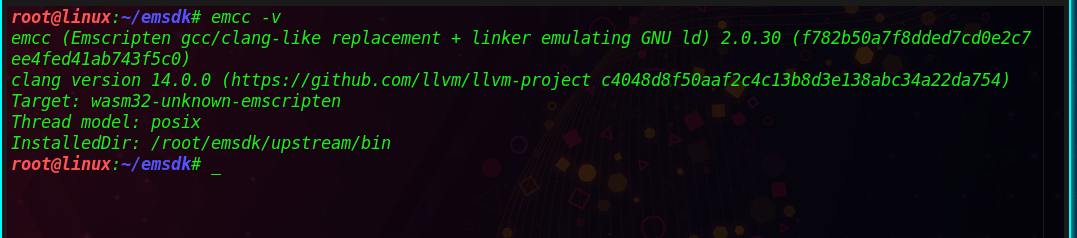
Figure 5:Successfully install emcc.

Figure 6:Hello world project.

Figure 7:Generated JavaScript output.

Figure 8:Generating HTML output.

Transpiling code base

A shell sort algorithm developed in c++ is transpiled in this case to native code.

Figure 9:Building code base.

Figure 10:Generated JavaScript.

Figure 11:Generating optimized HTML.

A simple python or PHP web server can be used to execute generated HTML code. Results can be viewed using a web browser.

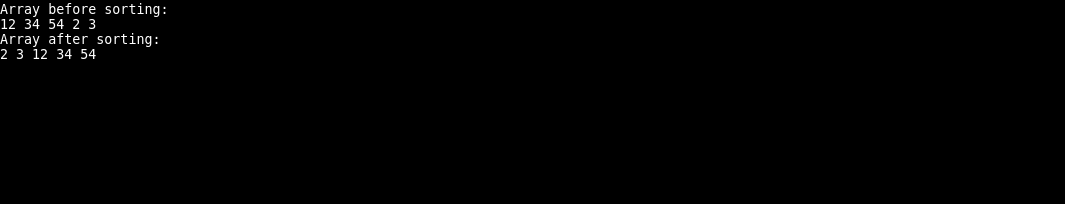


Figure 12:Using a web browser.

Performance Analysis

Code compiled on Web-Assembly runs at 1.55 times slower than native code. This is due to the following reasons:

* Code contained in Web-Assembly contains 2 times more loads and stores than native code.
* Web-Assembly requires more safety dynamic checks and therefore contains more branches than native code.

Appendix A

#include <iostream>

using namespace std;

/\* shellSort \*/

int shellSort(int arr[], int n)

{

for (int gap = n/2; gap > 0; gap /= 2)

{

for (int i = gap; i < n; i += 1)

{

int temp = arr[i];

int j;

for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)

arr[j] = arr[j - gap];

arr[j] = temp;

}

}

return 0;

}

void printArray(int arr[], int n)

{

for (int i=0; i<n; i++)

cout << arr[i] << " ";

}

int main()

{

int arr[] = {12, 34, 54, 2, 3}, i;

int n = sizeof(arr)/sizeof(arr[0]);

cout << "Array before sorting: \n";

printArray(arr, n);

shellSort(arr, n);

cout << "\nArray after sorting: \n";

printArray(arr, n);

return 0;

}