C Command Line Arguments Answers

Sayak Haldar

IIEST, Shibpur

1) Answer is b) Argument count, argument vector

2) Answer is a) int main(int var, char * varg[])

3) Answer is d) All of the mentioned

Explanation: Since all names given are valid executable file name

4) Answer is c) Jagged Array

Explanation: A jagged array is an array whose elements are arrays. The elements of a jagged array can be of different dimensions and sizes. A jagged array is sometimes called an "array of arrays.

5) Answer is d) Segmentation Fault

Explanation: argv[argc] will always result/cause segmentation fault.

(Though, the answer given in sanfoundry.com is a) (null). Does that mean the answer is compiler dependent. Since, in case of gcc compiler, argv[argc] will always cause segmentation fault

6. Answer is a) The number of command line arguments the program was invoked with

7) Answer is b) A pointer to an array of character strings that contain the arguments, one per string

Explanation: char *argv[] is an array of character pointers where each character pointer points to a string of character literals.

8) Answer is a) The name by which the program was invoked with

9) Answer is c) One

10) Answer is d) argc-1

11) Answer is b) 1 (the name of the executable file: The name by which the program was invoked with)

12) Answer is b) Executablefilename

Explanation: argc is post decremented at line 5 of the program

13) Answer is a) Segmentation fault/code crash

14) Answer is a) Segmentation fault/code crash

Explanation: argv is an array of character pointers. So, the name argv actually points to/represents the starting location of the array of character pointer. Now, focus on line 4 of the program.

while (*argv++ != NULL)

Now, *argv++ is actually *(argv++) Since post incrementation operator has higher precedence than indirection (Dereference) Operator. Now, since argv is post-incremented, it represents the starting address of the array of character pointers at line 4. So, *argv is argv[0] which is first character pointer of the array. Now, the first character pointer of the array points to the executable file name by which the program is invoked. So, control will enter the while loop (first time). Now, At line 5, argv represents the next address of the starting address of the array of character pointers. So, at line 5, *argv means argv[1], a character pointer which points to nothing. Now, %s in printf expects a character pointer. So, printf("%s\n", *argv) will actually try to dereference argv[1] to print the name pointed by it. It will cause segmentation fault since we cannot dereference a NULL pointer.

15) Answer is b) Executable file name

Explanation: argv is an array of character pointers. So, the name argv actually points to/represents the starting location of the array of character pointer. Now, focus on line 4 of the program.

while (*argv!= NULL)

Now, *argv means argv[0] which is a pointer points to the memory address which holds the executable file name as a string. So, *argv!= NULL this will return nonzero value (this checking condition will return nonzero value. Every relation operator returns some value in c by comparing the operands associated with it). So, control will enter the while loop (for first time or for first iteration)
Now, focus on line 5

printf("%s\n", *(argv++))

This will print the executable file name pointed by argv[0]. Now, notice that, argv is post incremented here. So, for the next iteration of while loop, argv will represent the next memory address of the starting memory address of the array of character pointers argv.

Now, again focus on the line 4 of the program for the next iteration of the while loop.

while (*argv!= NULL)

this will actually check if the pointer argv[1] points to Nothing (is a NULL pointer) or not. Now, argv[1] actually is a NULL pointer. So, for the second iteration of the while loop the validation condition (!= relational operator associated with two operands) returns zero value. So, the answer is program will only print the executable file name (due to the first iteration of the while loop).

16) Answer is a) Segmentation fault/code crash

Explanation: argv is an array of character pointers. So, the name argv actually points to/represents the starting location of the array of character pointer. Now, focus on line 4 of the program.

while (argv != NULL)

Now, argv represents the starting address of the array of character pointers. Now, an address can never by NULL. So, control enters the while loop for first iteration and print the executablefilename. Now, argv is post-incremented at line 5. So, in next iteration, argv will represent the next address of the starting address of the array of character pointers.

Now, for second iteration, argv represents the next address of the starting address of the array of character pointers. Now, at second iteration, line 5 will try to dereference a pointer will is a NULL pointer, which will cause segmentation fault/code crash.

References:

1. http://www.sanfoundry.com/c-interview-questions-answers/
2. http://stackoverflow.com/