

2622 2999 EdA 2 (2°C 2020)

Lea el texto "Difference between Static and Shared libraries" que su tutor le envió por mensajería de MleL. Luego elija la opción correcta.

Difference between Static and Shared libraries

In programming, a library is a collection of pre-compiled pieces of code that can be reused in a program. Libraries simplify life for programmers in that they provide reusable functions, routines, classes, data structures and so on which can be reused in the programs.

Static Libraries : A Static library or statically-linked library is a set of routines, external functions and variables which are resolved in a caller at compile-time and copied into a target application by a compiler, linker, or binder, producing an object file and a stand-alone executable. This executable and the process of compiling it are both known as a static build of the program. Historically, libraries could only be static. They are usually faster than the shared libraries because a set of commonly used object files is put into a single library executable file. One can build multiple executables without the need to recompile the file. Because it is a single file to be built, use of link commands is simpler than shared library link commands because you specify the name of the static library.

Shared Libraries: Shared libraries are .so (or in Windows .dll, or in OS X .dylib) files. These are linked dynamically simply including the address of the library (whereas static linking is a waste of space). Dynamic linking links the libraries at the run-time. Thus, all the functions are in a special place in memory space, and every program can access them, without having multiple copies of them.



Points: 0/20

1.La cadena léxica del primer párrafo "In programming, a library is a collection of pre-compiled pieces of code that can be reused in a program. Libraries simplify life for programmers, in that they provide reusable functions, routines, classes, data structures and so on which they can be reused in the programs." es: * (-/1 Points)

- ☐ a. In programming / pieces of code / in a program / life for programmers / in thatthey provide / so on / which / in the programs

- b. programming / library / collection / pre-compiled / pieces of code / can be ☒ reused / program / libraries / provide / reusable / functions / routines / classes / data structures / programs
- ☐ c. programming / is a / pieces / code / can be / program / simplify / programmers / in that / reusable / functions / so on / reused

2. En el fragmento "(...) copied into a target application (...)" **target:** *
(-/1 Points)

- ☐ a. Es un término transparente que significa objetivo
- ☐ b. Es un término transparente que significa tarjeta
- ☒ c. No es un término transparente

3. En el fragmento "(...) a set of commonly used object files (...)", **object:** *
(-/1 Points)

- ☐ a. Es un término transparente que significa objetivo
- ☒ b. Es un término transparente que significa objeto
- ☐ c. No es un término transparente

4. Lo contrario de "Thus, all the functions are in a special place in memory space, and every program can access them (...)" sería: *
(-/1 Points)

- ☐ a. Thus, all the functions are in a specific place in memory space, and each program can use them
- ☒ b. Thus, all the functions are in an indefinite place in memory space, and no program can use them

5. Los hipónimos del hiperónimo **library** son: * (-/1 Points)

- ☐ a. Functions, routines, classes, data structures
- ☐ b. Compiler, linker, binder
- ☒ c. Static library, shared library

6. La palabra **stand-alone** es un ejemplo de: * (-/1 Points)

- ☐ a. Prefijo
- ☐ b. Sufijo
- ☐ c. Prefijo y sufijo
- ☒ d. Palabra compuesta

7. La palabra **reusable** es un ejemplo de: * (-/1 Points)

- ☒ a. Prefijo
- ☐ b. Sufijo
- ☐ c. Prefijo y sufijo
- ☐ d. Palabra compuesta

8. La palabra **linker** es un ejemplo de: * (-/1 Points)

- ☐ a. Prefijo
- ☒ b. Sufijo
- ☐ c. Prefijo y sufijo
- ☐ d. Flexión

9. La palabra **faster** es un ejemplo de: * (-/1 Points)

- ☐ a. Prefijo
- ☐ b. Sufijo
- ☐ c. Prefijo y sufijo
- ☒ d. Flexión

10. La palabra **recompile** es un ejemplo de: * (-/1 Points)

- ☒ a. Prefijo
- ☐ b. Sufijo
- ☐ c. Prefijo y sufijo
- ☐ d. Flexión

11. En el fragmento "Dynamic linking links the libraries at the run-time" la palabra **links** es un ejemplo de flexión de: * (-/1 Points)

- ☐ a. Número
- ☐ b. Caso posesivo
- ☒ c. Tiempo verbal

12. En el fragmento "Dynamic linking links the libraries at the run-time" la palabra **libraries** es un ejemplo de flexión de: * (-/1 Points)

- ☒ a. Número
- ☐ b. Caso posesivo
- ☐ c. Tiempo verbal

13. En el fragmento "Dynamic linking links the libraries at the run-time" la palabra **linking** es un ejemplo de flexión -ING de: * (-/1 Points)

- ☐ a. Adjetivo
- ☒ b. Sustantivo
- ☐ c. Verbo

14. En la oración "This executable and the process of compiling it are both known as a static build of the program" las palabras estructurales son: * (-/1 Points)

- ☒ a. This / and / the / of / it / both / as / a / of / the
- ☐ b. Executable / process / compiling / are / known / static / build / program
- ☐ c. This / executable / the / process / compiling / it / known / as / static / build / the / program

15. En el fragmento "A Static library or statically-linked library is a set of routines, external functions and variables (...)" las palabras conceptuales son: * (-/1 Points)

- ☐ a. A / or / a / of / and
- ☐ b. A / static / library / statically-linked / library / a / set / routines / external / functions / variables
- ☒ c. Static / library / statically-linked / library / is / set / routines / external / functions / variables

16. En cuanto a las librerías / bibliotecas estáticas... * (-/1 Points)

- ☐ a. Uno debe recompilar el archivo una vez al construir cada ejecutable.
- ☐ b. Uno debe recompilar el archivo múltiples veces al construir cada ejecutable.
- ☒ c. Uno no necesita recompilar el archivo al construir múltiples ejecutables.

17.Traduzca la siguiente oración: "Because it is a single file to be built, use of link commands is simpler than shared library link commands because you specify the name of the static library." *

(-/2 Points)

Debido a que es un único archivo para compilar, el uso de comandos de enlace es más simple que usar los comandos de enlace de las bibliotecas compartidas, ya que vos especificás el nombre de la biblioteca estática.

18.Responda: ¿Qué beneficio trae el uso de las librerías / bibliotecas? *

(-/2 Points)

Ayuda a evitar el código duplicado, debido a que las bibliotecas contienen funciones, rutinas, estructuras de datos, etcétera, que pueden ser reutilizados en los programas.