



MODELO DE EXAMEN DEL PRIMER PARCIAL

Inglés Técnico (Universidad Nacional de La Matanza)



Scan to open on Studocu

LEA ATENTAMENTE EL TEXTO Y REALICE LA ACTIVIDAD A CONTINUACIÓN

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 are 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.

<https://www.geeksforgeeks.org/>

ELIJA LA OPCIÓN CORRECTA:

1. El texto “Difference between Static and Shared libraries” contiene:
 - a. contiene paratextos verbales.
 - b. contiene paratextos icónicos.
 - c. contiene paratextos verbales e icónicos.
2. En el fragmento “(...) copied into a **target** application (...)” target:
 - 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:
 - 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:
 - 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:
 - a. Functions, routines, classes, data structures
 - b. Compiler, linker, binder
 - c. Static library, shared library
6. El ejemplo de repetición léxica más prominente es:
 - a. Library
 - b. Function
 - c. Link
7. La palabra **stand-alone** es un ejemplo de:
 - a. Prefijo
 - b. Sufijo
 - c. Prefijo y sufijo
 - d. Palabra compuesta
8. La palabra **reusable** es un ejemplo de:
 - a. Prefijo
 - b. Sufijo
 - c. Prefijo y sufijo
 - d. Palabra compuesta
9. La palabra **linker** es un ejemplo de:
 - a. Prefijo
 - b. Sufijo
 - c. Prefijo y sufijo
 - d. Flexión
10. La palabra **faster** es un ejemplo de:
 - a. Prefijo
 - b. Sufijo
 - c. Prefijo y sufijo
 - d. Flexión
11. La palabra **recompile** en “es un ejemplo de:
 - a. Prefijo

- b. Sufijo
- c. Prefijo y sufijo
- d. Flexión

12. En el fragmento “*Dynamic linking **links** the libraries at the run-time*” la palabra **links** ¿qué tipo de ejemplo de flexión es? ¿Por qué?

13. En el fragmento “*Dynamic linking links the **libraries** at the run-time*” la palabra **libraries** es un ejemplo de flexión de:

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

14. En el fragmento “*Dynamic **linking** links the libraries at the run-time*” la palabra **linking** es un ejemplo de flexión -ING de:

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

15. 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:

- 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*

16. En el fragmento “*A Static library or statically-linked library is a set of routines, external functions and variables (...)*” las palabras conceptuales son:

- 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*

17. De acuerdo al texto el uso de las librerías / bibliotecas...

- a. facilita el trabajo a los programadores ya que las funciones que proveen pueden ser utilizadas varias veces.
- b. complica el trabajo a los programadores ya que las funciones que proveen pueden utilizarse solamente una vez.
- c. no proveen ningún beneficio a los programadores, ya que es lo mismo utilizarlas o no hacerlo.

18. En cuanto a las librerías / bibliotecas estáticas...
- 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.
19. El uso de los comandos de enlace...
- a. es más sencillo en las librerías / bibliotecas compartidas.
 - b. es más sencillo en las librerías / bibliotecas estáticas.
 - c. es igual de sencillo en ambos casos.
20. Traduzca el siguiente fragmento:
- “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..”

TOTAL SCORE: 20 points

PASSING SCORE: 10 /20
