Builds deterministas en C++

Carlos Zoido
Ingeniero Industrial
Conan.io developer | **JFrog** 🐸
@czoido

¿Qué son?

Código Fuente

Código Fuente

Toolchain

Código Fuente

Toolchain

Binario


C++
CPP
101
0011
1010
BIN

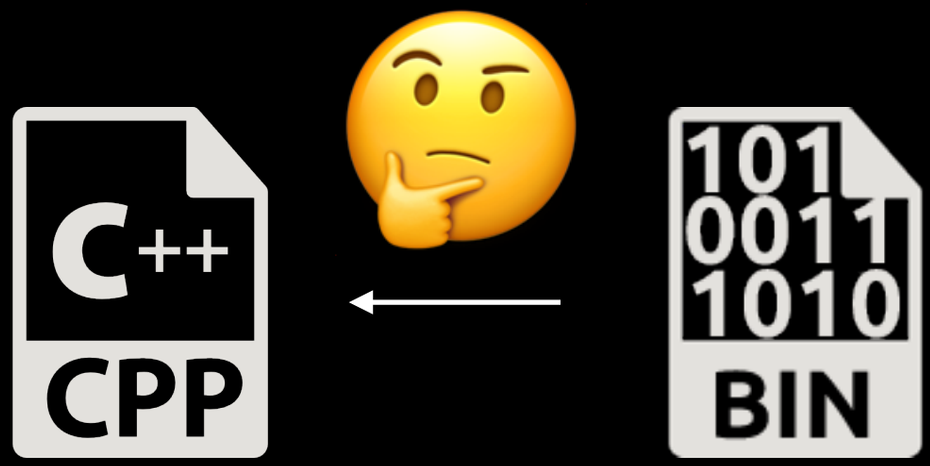

C++
CPP
101
0011
1010
BIN


C++
CPP
101
0011
1010
BIN


C++
CPP
101
0011
1010
BIN


C++
CPP
101
0011
1010
BIN


C++
CPP
101
0011
1010
BIN


C++
CPP
101
0011
1010
BIN

¿Por qué son importantes?

Código Fuente

Toolchain

Binario

“Verificable”

“Usable”

Código Fuente

Toolchain

Binario

"Verificable"

"Usable"

‘OpenSSH Off-By-One Vulnerability’

Published on March 7th, 2002

Summary

‘A security bug in OpenSSH that can be exploited locally by an authenticated user logging into a vulnerable OpenSSH server or by a malicious SSH server attacking a vulnerable OpenSSH client allows them to create a buffer overflow attack.’

Credit:

‘The information has been provided by Joost Pol.’

Details

‘**Systems affected:**
All versions of OpenSSH between 2.0 and 3.0.2 contain an off-by-one error in the channel code.

Immune systems:
OpenSSH 3.1 and later are not affected.

Solution:
Upgrade to OpenSSH 3.1 or apply the following patch.

Patch:

```
Index: channels.c
===================================================================
RCS file: /cvs/src/usr.bin/ssh/channels.c,v
retrieving revision 1.170
retrieving revision 1.171
diff -u -r1.170 -r1.171
— channels.c 27 Feb 2002 21:23:13 -0000 1.170
+++ channels.c 4 Mar 2002 19:37:58 -0000 1.171
@@ -146,7 +146,7 @@
```

– channels.c 27 Feb 2002 21:23:13 -0000 1.170

+++ channels.c 4 Mar 2002 19:37:58 -0000 1.171

@@ -146,7 +146,7 @@

{

Channel *c;

– if (id < 0 || id > channels_alloc) {

JLE (<) | 7C

JL (<=) | 7E

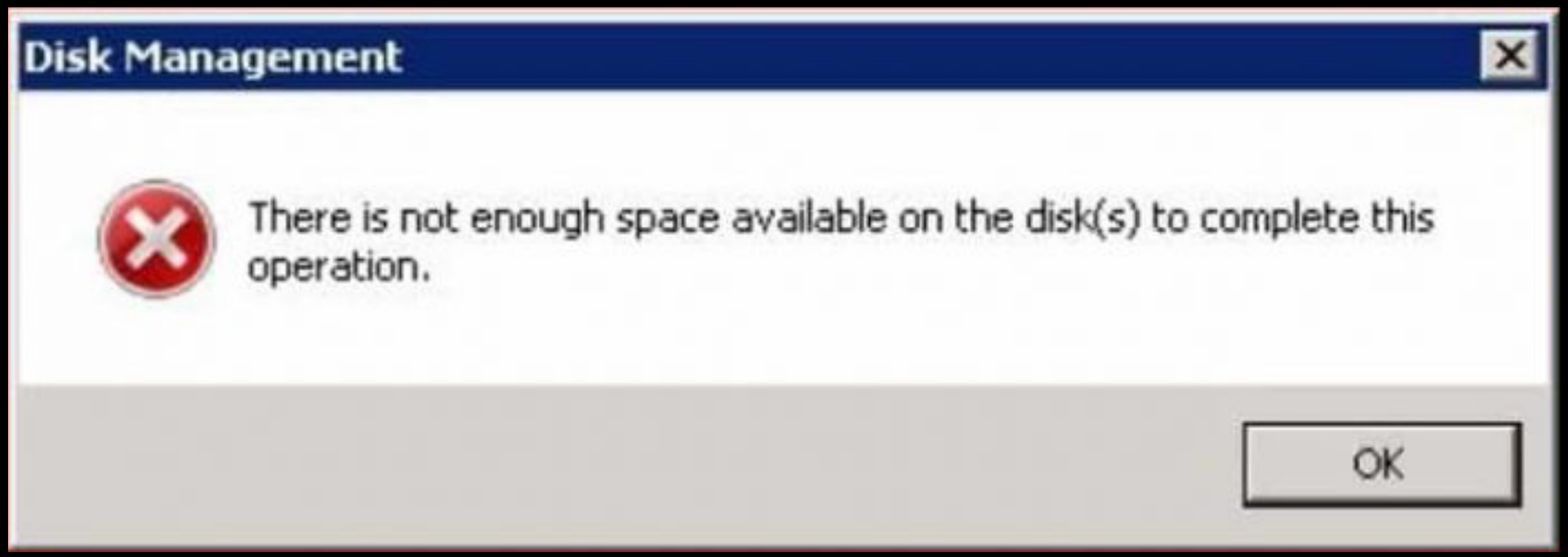
+ if (id < 0 || id >= channels_alloc) {

log(‘channel_lookup: %d: bad id’, id);

return NULL;

}’

XCodeGhost


Disk Management
There is not enough space available on the disk(s) to complete this operation.
OK

Causas

Windows (PE)	Linux (ELF)	MacOS (Mach-O)
.obj	.o	.o
.lib	.a	.a
.dll	.so	.dylib
.exe	none	none
dumpbin	readelf	otool

1. **Timestamps**: código fuente o toolchain
2. Información de **carpetas** propagada a los binarios
3. **Orden** de los archivos proporcionados al build system
4. **Aleatoriedad** en el compilador

Timestamps

```
.
├── CMakeLists.txt
├── hello_world.cpp
├── hello_world.hpp
└── main.cpp
```

```cpp
#include "hello_world.hpp"

#include <iostream>

void HelloWorld::PrintMessage(const std::string & message)
{
   std::cout << message << std::endl;
}
```

```
.
├── CMakeLists.txt
├── hello_world.cpp
├── hello_world.hpp
└── main.cpp
```

```cpp
#include <iostream>
#include "hello_world.hpp"

int main(int argc, char** argv)
{
    HelloWorld hello;
    hello.PrintMessage("Hello World!");
    return 0;
}
```

```
cmake_minimum_required(VERSION 3.0)
project(HelloWorld)
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED ON)
add_library(HelloLibA hello_world.cpp)
add_library(HelloLibB hello_world.cpp)
add_executable(helloA main.cpp)
add_executable(helloB main.cpp)
target_link_libraries(helloA HelloLibA)
target_link_libraries(helloB HelloLibB)
```

```
cd build
cmake ..
make
md5sum CMakeFiles/HelloLibA.dir/hello_world.cpp.o
md5sum CMakeFiles/HelloLibB.dir/hello_world.cpp.o
md5sum libHelloLibA.a
md5sum libHelloLibB.a
md5sum helloA
md5sum helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents…
Scanning dependencies of target HelloLibA
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
Scanning dependencies of target helloA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
Scanning dependencies of target HelloLibB
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
Scanning dependencies of target helloB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
0bdcda17e50a3db5b7ac0fea4b9122e7  libHelloLibA.a
f4f192dde17dfaaa905747003718 47ab  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents…
Scanning dependencies of target HelloLibA
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
Scanning dependencies of target helloA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
Scanning dependencies of target HelloLibB
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
Scanning dependencies of target helloB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
0bdcda17e50a3db5b7ac0fea4b9122e7  libHelloLibA.a
f4f192dde17dfaaa905747003718 47ab  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents…
Scanning dependencies of target HelloLibA
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
Scanning dependencies of target helloA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
Scanning dependencies of target HelloLibB
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
Scanning dependencies of target helloB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
❌ 0bdcda17e50a3db5b7ac0fea4b9122e7  libHelloLibA.a
f4f192dde17dfaaa905747003718 47ab  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents…
Scanning dependencies of target HelloLibA
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
Scanning dependencies of target helloA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
Scanning dependencies of target HelloLibB
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
Scanning dependencies of target helloB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
❌ 0bdcda17e50a3db5b7ac0fea4b9122e7  libHelloLibA.a
❌ f4f192dde17dfaaa905747003718 47ab  libHelloLibB.a
✅ 61c25ff1d364cb1190b14ee75564fea4  helloA
✅ 61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  build otool -a libHelloLibA.a
Archive : libHelloLibA.a
0100644 503/20    612 1571672638 #1/20
0100644 503/20  13036 1571672633 #1/28
➜  build otool -a libHelloLibB.a
Archive : libHelloLibB.a
0100644 503/20    612 1571672639 #1/20
0100644 503/20  13036 1571672634 #1/28
```

```
➜  build otool -a libHelloLibA.a
Archive : libHelloLibA.a
0100644 503/20    612 1571672638 #1/20
0100644 503/20  13036 1571672633 #1/28
➜  build otool -a libHelloLibB.a
Archive : libHelloLibB.a
0100644 503/20    612 1571672639 #1/20
0100644 503/20  13036 1571672634 #1/28
```

Discussion

A static archive library begins with the file identifier string `!<arch>`, followed by a newline character (ASCII value 0x0A). The file identifier string is followed by a series of member files. Each member consists of a fixed-length header line followed by the file data. The header line is 60 bytes long and is divided into five fixed-length fields, as shown in this example header line:

```
grapple.c       999514211   501   20    100644  167       `
```

The last 2 bytes of the header line are a grave accent (`) character (ASCII value 0x60) and a newline character. All header fields are defined in ASCII and padded with spaces to the full length of the field. All fields are defined in decimal notation, except for the file mode field, which is defined in octal. These are the descriptions for each field:

- The name field (16 bytes) contains the name of the file. If the name is either longer than 16 bytes or contains a space character, the actual name should be written directly after the header line and the name field should contain the string `#1/` followed by the length. To keep the archive entries aligned to 4 byte boundaries, the length of the name that follows the `#1/` is rounded to 4 bytes and the name that follows the header is padded with null bytes.
- The modified date field (12 bytes) is taken from the `st_time` field returned by the `stat` system call.
- The user ID field (6 bytes) is taken from the `st_uid` field returned by the `stat` system call.
- The group ID field (6 bytes) is taken from the `st_gid` field returned by the `stat` system call.
- The file mode field (8 bytes) is taken from the `st_mode` field returned by the `stat` system call. This field is written in octal notation.
- The file size field (8 bytes) is taken from the `st_size` field returned by the `stat` system call.

The first member in a static archive library is always the symbol table describing the contents of the rest of the member files. This member is always called either `__.SYMDEF` or `__.SYMDEF SORTED` (note the two leading underscores and the period). The name used depends on the sort order of the symbol table. The older variant—`__.SYMDEF`—contains entries in the same order that they appear in the object files. The newer variant—`__.SYMDEF SORTED`— contains entries in alphabetical order, which allows the static linker to load the symbols faster.

The `__.SYMDEF` and `.__SORTED SYMDEF` archive members contain an array of `ranlib` data structures preceded by the length in bytes (a long integer, 4 bytes) of the number of items in the array. The array is followed by a string table of null-terminated strings, which are preceded by the length in bytes of the entire string table (again, a 4-byte long integer).

The string table is an array of C strings, each terminated by a null byte.

The `ranlib` declarations can be found in `/usr/include/mach-o/ranlib.h`.

Discussion

A static archive library begins with the file identifier string `!<arch>`, followed by a newline character (ASCII value 0x0A). The file identifier string is followed by a series of member files. Each member consists of a fixed-length header line followed by the file data. The header line is 60 bytes long and is divided into five fixed-length fields, as shown in this example header line:

```
grapple.c       999514211   501   20    100644  167       `
```

The last 2 bytes of the header line are a grave accent (`) character (ASCII value 0x60) and a newline character. All header fields are defined in ASCII and padded with spaces to the full length of the field. All fields are defined in decimal notation, except for the file mode field, which is defined in octal. These are the descriptions for each field:

- The name field (16 bytes) contains the name of the file. If the name is either longer than 16 bytes or contains a space character, the actual name should be written directly after the header line and the name field should contain the string `#1/` followed by the length. To keep the archive entries aligned to 4 byte boundaries, the length of the name that follows the `#1/` is rounded to 4 bytes and the name that follows the header is padded with null bytes.
- The modified date field (12 bytes) is taken from the `st_time` field returned by the `stat` system call.
- The user ID field (6 bytes) is taken from the `st_uid` field returned by the `stat` system call.
- The group ID field (6 bytes) is taken from the `st_gid` field returned by the `stat` system call.
- The file mode field (8 bytes) is taken from the `st_mode` field returned by the `stat` system call. This field is written in octal notation.
- The file size field (8 bytes) is taken from the `st_size` field returned by the `stat` system call.

The first member in a static archive library is always the symbol table describing the contents of the rest of the member files. This member is always called either `__.SYMDEF` or `__.SYMDEF SORTED` (note the two leading underscores and the period). The name used depends on the sort order of the symbol table. The older variant—`__.SYMDEF`—contains entries in the same order that they appear in the object files. The newer variant—`__.SYMDEF SORTED`— contains entries in alphabetical order, which allows the static linker to load the symbols faster.

The `__.SYMDEF` and `.__SORTED SYMDEF` archive members contain an array of `ranlib` data structures preceded by the length in bytes (a long integer, 4 bytes) of the number of items in the array. The array is followed by a string table of null-terminated strings, which are preceded by the length in bytes of the entire string table (again, a 4-byte long integer).

The string table is an array of C strings, each terminated by a null byte.

The `ranlib` declarations can be found in `/usr/include/mach-o/ranlib.h`.

```cpp
#include <iostream>
#include "hello_world.hpp"

int main(int argc, char** argv)
{
    HelloWorld hello;
    hello.PrintMessage("Hello World!");
    std::cout << "At time: " << __TIME__ << std::endl;
    return 0;
}
```

```cpp
#include <iostream>
#include "hello_world.hpp"

int main(int argc, char** argv)
{
    HelloWorld hello;
    hello.PrintMessage("Hello World!");
    std::cout << "At time: " << __TIME__ << std::endl;
    return 0;
}
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
efdedd6cecc02a6775ca6d0054ac5b78  libHelloLibA.a
20f56367227508ceccd431701f07a151  libHelloLibB.a
c9a724d137a65cfafe77f0109881059e  helloA
67b814ae0028b0e25ef4303b52c8a3d0  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
efdedd6cecc02a6775ca6d0054ac5b78  libHelloLibA.a
20f56367227508ceccd431701f07a151  libHelloLibB.a
c9a724d137a65cfafe77f0109881059e  helloA
67b814ae0028b0e25ef4303b52c8a3d0  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
❌ efdedd6cecc02a6775ca6d0054ac5b78  libHelloLibA.a
20f56367227508ceccd431701f07a151  libHelloLibB.a
c9a724d137a65cfafe77f0109881059e  helloA
67b814ae0028b0e25ef4303b52c8a3d0  helloB
```

```
➜  hello_world ./run_build.sh
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
❌ efdedd6cecc02a6775ca6d0054ac5b78  libHelloLibA.a
❌ 20f56367227508ceccd431701f07a151  libHelloLibB.a
❌ c9a724d137a65cfafe77f0109881059e  helloA
❌ 67b814ae0028b0e25ef4303b52c8a3d0  helloB
```

```
--- helloA
+++ helloB
├── otool -arch x86_64 -tdvV {}
|--- Code for architecture x86_64
| @@ -16,15 +16,15 @@
|  0000000100000 18da jmp   0x100000 18df
|  0000000100000 18df leaq  -0x30(%rbp), %rdi
|  0000000100000 18e3 callq 0x100002d54 ## symbol stub for:
__ZNSt3__112basic_stringIcNS_11char_traitsIcEENS_9allocatorIcEEED1Ev
|  0000000100000 18e8 movq  0x1719(%rip), %rdi ## literal pool symbol address: __ZNSt3__14coutE
|  0000000100000 18ef leaq  0x162f(%rip), %rsi ## literal pool for: "At time: "
|  0000000100000 18f6 callq 0x100002d8a ## symbol stub for:
  ZNSt3  1lsINS 11char traitsIcEEEERNS 13basic ostreamIcT EES6 PKc
|  0000000100000 18fb movq  %rax, %rdi
| -0000000100000 18fe leaq  0x162a(%rip), %rsi ## literal pool for: "17:56:38"
| +0000000100000 18fe leaq  0x162a(%rip), %rsi ## literal pool for: "17:56:39"
|  0000000100001905 callq 0x100002d8a ## symbol stub for:
__ZNSt3__1lsINS_11char_traitsIcEEEERNS_13basic_ostreamIcT_EES6_PKc
|  0000000100001 90a movq  %rax, %rdi
|  0000000100001 90d leaq  __ZNSt3__1L4endlIcNS_11char_traitsIcEEEERNS_13basic_ostreamIT_T0_EES7_(%rip),
%rsi ## std::__1::basic_ostream<char, std::__1::char_traits<char> >& std::__1::endl<char,
std::__1::char_traits<char> >(std::__1::basic_ostream<char, std::__1::char_traits<char> >&)
|  0000000100001914 callq __ZNSt3__113basic_ostreamIcNS_11char_traitsIcEEElsEPFRS3_S4_E ##
std::__1::basic_ostream<char, std::__1::char_traits<char> >::operator<<(std::__1::basic_ostream<char,
std::__1::char_traits<char> >& (*)(std::__1::basic_ostream<char, std::__1::char_traits<char> >&))
|  0000000100001919 xorl  %ecx, %ecx
|  0000000100001 91b movq  %rax, -0x50(%rbp)
|  0000000100001 91f movl  %ecx, %eax
```

```
cofiCEEEDiEv
bool symbol address: __ZNSt3_
bool for: "At time: "
:
_EES6_PKc
bool for: "17:56:38"
bool for: "17:56:39"
:
_EES6_PKc

aitsIcEEEERNS_13basic_ostream
```

Microsoft Visual Studio: flag del linker sin documentar: **/Brepro**

gcc: **SOURCE_DATE_EPOCH** arregla las macros __DATE__ y __TIME___

clang: **ZERO_AR_DATE** arregla los timestamps del formato archive pero no las macros __DATE__ y __TIME___

```
➜  hello_world export ZERO_AR_DATE=1

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer/…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibA.a
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world export ZERO_AR_DATE=1 ←

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer/…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibA.a
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world export ZERO_AR_DATE=1 ←

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer/…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibA.a
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world export ZERO_AR_DATE=1 ←

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer/…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
✅ 2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibA.a
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibB.a
61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

```
➜  hello_world export ZERO_AR_DATE=1 ←

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer/…
[ 12%] Building CXX object CMakeFiles/HelloLibA.dir/hello_world.cpp.o
[ 25%] Linking CXX static library libHelloLibA.a
[ 25%] Built target HelloLibA
[ 37%] Building CXX object CMakeFiles/helloA.dir/main.cpp.o
[ 50%] Linking CXX executable helloA
[ 50%] Built target helloA
[ 62%] Building CXX object CMakeFiles/HelloLibB.dir/hello_world.cpp.o
[ 75%] Linking CXX static library libHelloLibB.a
[ 75%] Built target HelloLibB
[ 87%] Building CXX object CMakeFiles/helloB.dir/main.cpp.o
[100%] Linking CXX executable helloB
[100%] Built target helloB
✅ 1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibA.dir/hello_world.cpp.o
1157d509c31b2df484181b6d94b61203  CMakeFiles/HelloLibB.dir/hello_world.cpp.o
✅ 2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibA.a
2813f35dc4c0cc24d60ed2769ce52eb0  libHelloLibB.a
✅ 61c25ff1d364cb1190b14ee75564fea4  helloA
61c25ff1d364cb1190b14ee75564fea4  helloB
```

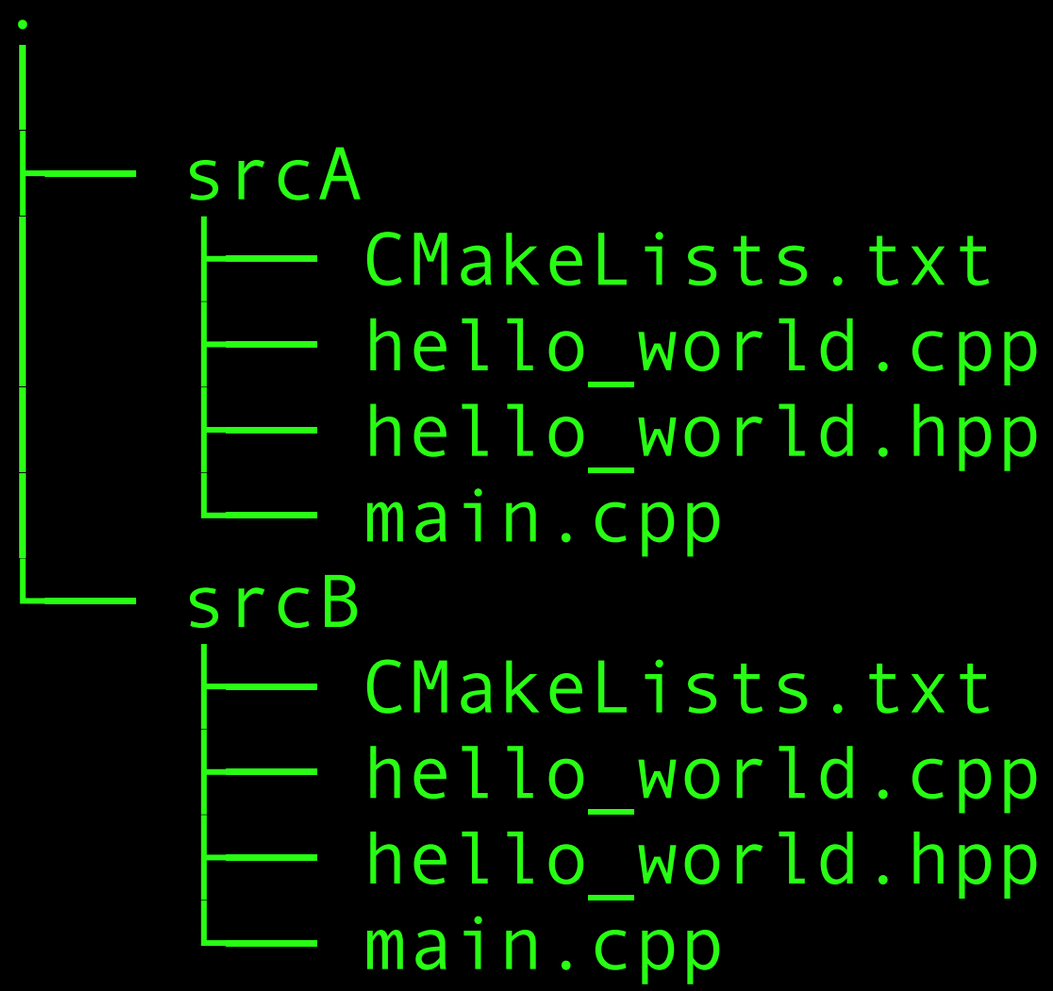
```
➜  build otool -a libHelloLibA.a
Archive : libHelloLibA.a
0100644 503/20     612 0 #1/20
0100644 503/20   13036 0 #1/28
➜  build otool -a libHelloLibB.a
Archive : libHelloLibB.a
0100644 503/20     612 0 #1/20
0100644 503/20   13036 0 #1/28
```

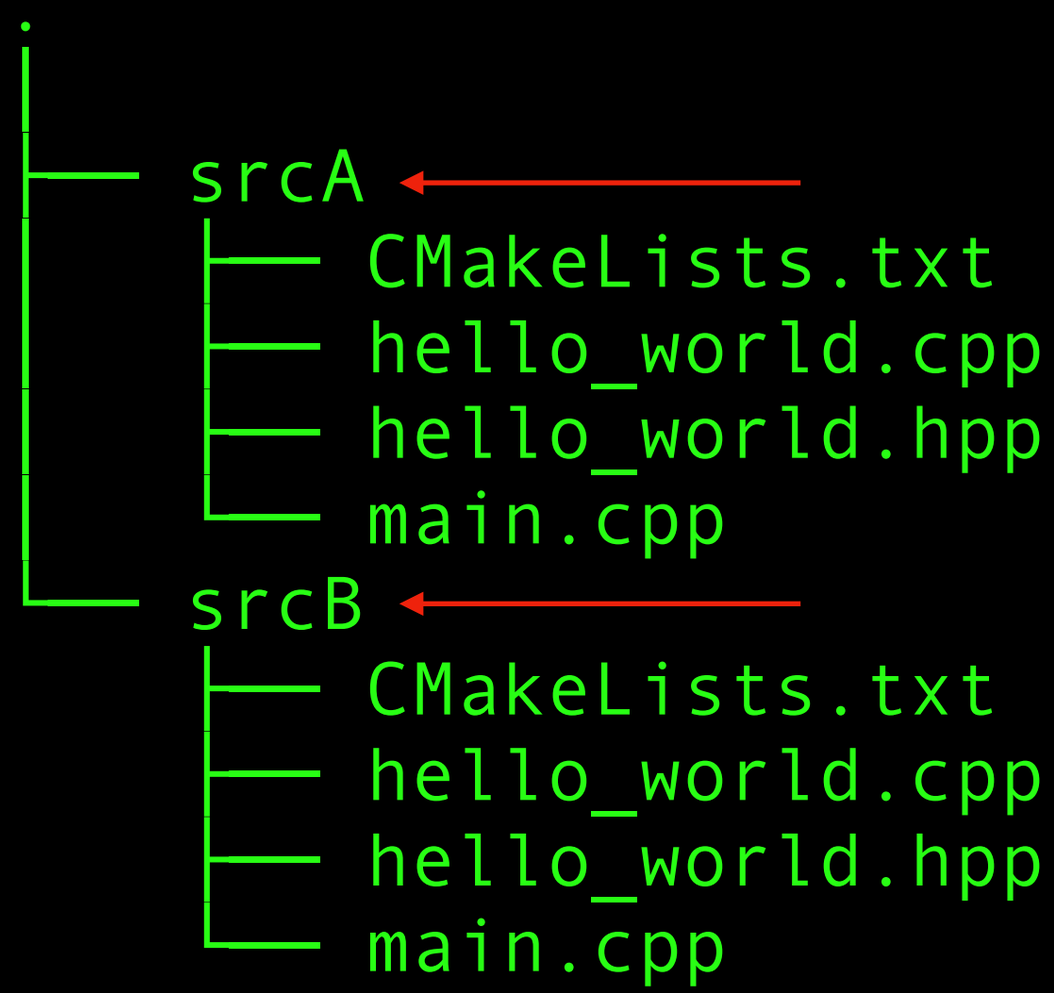
```
➜  build otool -a libHelloLibA.a
Archive : libHelloLibA.a
0100644 503/20     612 0 #1/20
0100644 503/20   13036 0 #1/28
➜  build otool -a libHelloLibB.a
Archive : libHelloLibB.a
0100644 503/20     612 0 #1/20
0100644 503/20   13036 0 #1/28
```

Información de carpetas

Mismo código fuente compilado en dos carpetas diferentes

- Uso de la macro **__FILE__**
- Compilando binarios en modo **DEBUG** que guardan información de dónde está el código fuente

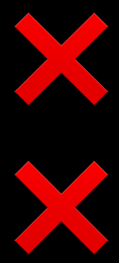

.
srcA
CMakeLists.txt
hello_world.cpp
hello_world.hpp
main.cpp
srcB
CMakeLists.txt
hello_world.cpp
hello_world.hpp
main.cpp


.
srcA
CMakeLists.txt
hello_world.cpp
hello_world.hpp
main.cpp
srcB
CMakeLists.txt
hello_world.cpp
hello_world.hpp
main.cpp

```
cd srcA/build
cmake -DCMAKE_BUILD_TYPE=Debug ..
make clean
make
cd .. && cd ..
cd srcB/build
cmake -DCMAKE_BUILD_TYPE=Debug ..
make clean
make
cd .. && cd ..
md5sum srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
md5sum srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
md5sum srcA/build/libHelloLib.a
md5sum srcB/build/libHelloLib.a
md5sum srcA/build/hello
md5sum srcB/build/hello
diffoscope srcA/build/hello srcB/build/hello
diffoscope srcA/build/libHelloLib.a srcB/build/libHelloLib.a
diffoscope …/hello_world.cpp.o …/hello_world.cpp.o
```

```
49d62e816e1418e94a93ba0194ae2668  hello_world.cpp.o
4f33b655b1797c0afc2122595666abf9  hello_world.cpp.o
b172d25f4e2896d935229f4099e07fcd  srcA/build/libHelloLib.a
fc60b646e2bc112a7cdce8cbd6a55a3f  srcB/build/libHelloLib.a
f9819ffb851aaa71cb1e4bf8cdf7a7b1  srcA/build/hello
588354e0cfc6d3cec1699b0e71237a9b  srcB/build/hello
```

```
✗ 49d62e816e1418e94a93ba0194ae2668  hello_world.cpp.o
  4f33b655b1797c0afc2122595666abf9  hello_world.cpp.o
  b172d25f4e2896d935229f4099e07fcd  srcA/build/libHelloLib.a
  fc60b646e2bc112a7cdce8cbd6a55a3f  srcB/build/libHelloLib.a
  f9819ffb851aaa71cb1e4bf8cdf7a7b1  srcA/build/hello
  588354e0cfc6d3cec1699b0e71237a9b  srcB/build/hello
```



```
49d62e816e1418e94a93ba0194ae2668  hello_world.cpp.o
4f33b655b1797c0afc2122595666abf9  hello_world.cpp.o
b172d25f4e2896d935229f4099e07fcd  srcA/build/libHelloLib.a
fc60b646e2bc112a7cdce8cbd6a55a3f  srcB/build/libHelloLib.a
f9819ffb851aaa71cb1e4bf8cdf7a7b1  srcA/build/hello
588354e0cfc6d3cec1699b0e71237a9b  srcB/build/hello
```

```
✗ 49d62e816e1418e94a93ba0194ae2668  hello_world.cpp.o
4f33b655b1797c0afc2122595666abf9  hello_world.cpp.o
✗ b172d25f4e2896d935229f4099e07fcd  srcA/build/libHelloLib.a
fc60b646e2bc112a7cdce8cbd6a55a3f  srcB/build/libHelloLib.a
✗ f9819ffb851aaa71cb1e4bf8cdf7a7b1  srcA/build/hello
✗ 588354e0cfc6d3cec1699b0e71237a9b  srcB/build/hello
```

```
588354e0cfc6d3cec1699b0e71237a9b  srcB/build/hello
2019-10-21 18:26:31 W: diffoscope.main: Fuzzy-matching is currently disabled as the "tlsh"
module is unavailable.
--- srcA/build/hello
+++ srcB/build/hello
|--- Format-specific differences are supported for MacOS binaries but no file-specific
differences were detected; falling back to a binary diff. file(1) reports: Mach-O 64-bit
x86_64 executable, flags:<NOUNDEFS|DYLDLINK|TWOLEVEL|WEAK_DEFINES|BINDS_TO_WEAK|PIE>
@@ -1281,21 +1281,21 @@
 00005000: 5f76 3000 5f6d 656d 7365 7400 5f73 7472  _v0._memset._str
 00005010: 6c65 6e00 6479 6c64 5f73 7475 625f 6269  len.dyld_stub_bi
 00005020: 6e64 6572 002f 5573 6572 732f 6361 726c  nder./Users/carl
 00005030: 6f73 2f44 6f63 756d 656e 7473 2f64 6576  os/Documents/dev
 00005040: 656c 6f70 6572 2f72 6570 726f 6475 6369  eloper/reproduci
 00005050: 626c 652d 6275 696c 6473 2f73 616e 6462  ble-builds/sandb
 00005060: 6f78 2f68 656c 6c6f 5f77 6f72 6c64 5f64  ox/hello_world_d
-00005070: 6562 7567 2f73 7263 412f 006d 6169 6e2e  ebug/srcA/.main.
+00005070: 6562 7567 2f73 7263 422f 006d 6169 6e2e  ebug/srcB/.main.
 00005080: 6370 7000 2f55 7365 7273 2f63 6172 6c6f  cpp./Users/carlo
 00005090: 732f 446f 6375 6d65 6e74 732f 6465 7665  s/Documents/deve
 000050a0: 6c6f 7065 722f 7265 7072 6f64 7563 6962  loper/reproducib
 000050b0: 6c65 2d62 7569 6c64 732f 7361 6e64 626f  le-builds/sandbo
 000050c0: 782f 6865 6c6c 6f5f 776f 726c 645f 6465  x/hello_world_de
-000050d0: 6275 672f 7372 6341 2f62 7569 6c64 2f43  bug/srcA/build/C
+000050d0: 6275 672f 7372 6342 2f62 7569 6c64 2f43  bug/srcB/build/C
 000050e0: 4d61 6b65 4669 6c65 732f 6865 6c6c 6f2e  MakeFiles/hello.
 000050f0: 6469 722f 6d61 696e 2e63 7070 2e6f 005f  dir/main.cpp.o._
 00005100: 6d61 696e 005f 5f5a 4e53 7433 5f5f 3131  main.__ZNSt3__11
 00005110: 3262 6173 6963 5f73 7472 696e 6749 634e  2basic_stringIcN
 00005120: 535f 3131 6368 6172 5f74 7261 6974 7349  S_11char_traitsI
 00005130: 6345 454e 535f 3961 6c6c 6f63 6174 6f72  cEENS_9allocator
 00005140: 4963 4545 4543 3149 446e 4545 504b 6300  IcEEEC1IDnEEPKc.
@@ -1336,15 +1336,15 @@
 00005370: 6578 6365 7074 5f74 6162 6c65 3000 6865  except_table0.he
 00005380: 6c6c 6f5f 776f 726c 642e 6370 7000 2f55  llo_world.cpp./U
 00005390: 7365 7273 2f63 6172 6c6f 732f 446f 6375  sers/carlos/Docu
 000053a0: 6d65 6e74 732f 6465 7665 6c6f 7065 722f  ments/developer/
 000053b0: 7265 7072 6f64 7563 6962 6c65 2d62 7569  reproducible-bui
```

```
 00005450: 6974 7349 6345 454e 5330 5f39 616c 6c6f  itsIcEENS0_9allo
2019-10-21 18:26:32 W: diffoscope.main: Fuzzy-matching is currently disabled as the "tlsh" module is unavailable.
--- srcA/build/libHelloLib.a
+++ srcB/build/libHelloLib.a
|--- 'readelf' not available in path. Falling back to binary comparison.
@@ -347,21 +347,21 @@
 000015a0: 616e 6720 7665 7273 696f 6e20 3131 2e30  ang version 11.0
 000015b0: 2e30 2028 636c 616e 672d 3131 3030 2e30  .0 (clang-1100.0
 000015c0: 2e33 332e 3829 002f 5573 6572 732f 6361  .33.8)./Users/ca
 000015d0: 726c 6f73 2f44 6f63 756d 656e 7473 2f64  rlos/Documents/d
 000015e0: 6576 656c 6f70 6572 2f72 6570 726f 6475  eveloper/reprodu
 000015f0: 6369 626c 652d 6275 696c 6473 2f73 616e  cible-builds/san
 00001600: 6462 6f78 2f68 656c 6c6f 5f77 6f72 6c64  dbox/hello_world
-00001610: 5f64 6562 7567 2f73 7263 412f 6865 6c6c  _debug/srcA/hell
+00001610: 5f64 6562 7567 2f73 7263 422f 6865 6c6c  _debug/srcB/hell
 00001620: 6f5f 776f 726c 642e 6370 7000 2f55 7365  o_world.cpp./Use
 00001630: 7273 2f63 6172 6c6f 732f 446f 6375 6d65  rs/carlos/Docume
 00001640: 6e74 732f 6465 7665 6c6f 7065 722f 7265  nts/developer/re
 00001650: 7072 6f64 7563 6962 6c65 2d62 7569 6c64  producible-build
 00001660: 732f 7361 6e64 626f 782f 6865 6c6c 6f5f  s/sandbox/hello_
-00001670: 776f 726c 645f 6465 6275 672f 7372 6341  world_debug/srcA
+00001670: 776f 726c 645f 6465 6275 672f 7372 6342  world_debug/srcB
 00001680: 2f62 7569 6c64 0073 7464 005f 5f31 0069  /build.std.__1.i
 00001690: 6f73 5f62 6173 6500 5f76 7074 7224 696f  os_base._vptr$io
 000016a0: 735f 6261 7365 005f 5f76 7462 6c5f 7074  s_base.__vtbl_pt
 000016b0: 725f 7479 7065 0069 6e74 0062 6f6f 6c61  r_type.int.boola
 000016c0: 6c70 6861 0066 6d74 666c 6167 7300 756e  lpha.fmtflags.un
 000016d0: 7369 676e 6564 2069 6e74 0064 6563 0066  signed int.dec.f
 000016e0: 6978 6564 0068 6578 0069 6e74 6572 6e61  ixed.hex.interna
@@ -5730,15 +5730,15 @@
 00016610: 2f69 6e63 6c75 6465 2f6d 616c 6c6f 6300  /include/malloc.
 00016620: 2f75 7372 2f69 6e63 6c75 6465 2f73 7973  /usr/include/sys
 00016630: 002f 5573 6572 732f 6361 726c 6f73 2f44  ./Users/carlos/D
 00016640: 6f63 756d 656e 7473 2f64 6576 656c 6f70  ocuments/develop
 00016650: 6572 2f72 6570 726f 6475 6369 626c 652d  er/reproducible-
 00016660: 6275 696c 6473 2f73 616e 6462 6f78 2f68  builds/sandbox/h
 00016670: 656c 6c6f 5f77 6f72 6c64 5f64 6562 7567  ello_world_debug
```

```
2019-10-21 18:26:32 W: diffoscope.main: Fuzzy-matching is currently disabled as the "tlsh"
module is unavailable.
--- srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
+++ srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
|--- Format-specific differences are supported for MacOS binaries but no file-specific
differences were detected; falling back to a binary diff. file(1) reports: Mach-O 64-bit
x86_64 object, flags:<|SUBSECTIONS_VIA_SYMBOLS>
@@ -299,21 +299,21 @@
 000012a0: 616e 6720 7665 7273 696f 6e20 3131 2e30  ang version 11.0
 000012b0: 2e30 2028 636c 616e 672d 3131 3030 2e30  .0 (clang-1100.0
 000012c0: 2e33 332e 3829 002f 5573 6572 732f 6361  .33.8)./Users/ca
 000012d0: 726c 6f73 2f44 6f63 756d 656e 7473 2f64  rlos/Documents/d
 000012e0: 6576 656c 6f70 6572 2f72 6570 726f 6475  eveloper/reprodu
 000012f0: 6369 626c 652d 6275 696c 6473 2f73 616e  cible-builds/san
 00001300: 6462 6f78 2f68 656c 6c6f 5f77 6f72 6c64  dbox/hello_world
-00001310: 5f64 6562 7567 2f73 7263 412f 6865 6c6c  _debug/srcA/hell
+00001310: 5f64 6562 7567 2f73 7263 422f 6865 6c6c  _debug/srcB/hell
 00001320: 6f5f 776f 726c 642e 6370 7000 2f55 7365  o_world.cpp./Use
 00001330: 7273 2f63 6172 6c6f 732f 446f 6375 6d65  rs/carlos/Docume
 00001340: 6e74 732f 6465 7665 6c6f 7065 722f 7265  nts/developer/re
 00001350: 7072 6f64 7563 6962 6c65 2d62 7569 6c64  producible-build
 00001360: 732f 7361 6e64 626f 782f 6865 6c6c 6f5f  s/sandbox/hello_
-00001370: 776f 726c 645f 6465 6275 672f 7372 6341  world_debug/srcA
+00001370: 776f 726c 645f 6465 6275 672f 7372 6342  world_debug/srcB
 00001380: 2f62 7569 6c64 0073 7464 005f 5f31 0069  /build.std.__1.i
 00001390: 6f73 5f62 6173 6500 5f76 7074 7224 696f  os_base._vptr$io
 000013a0: 735f 6261 7365 005f 5f76 7462 6c5f 7074  s_base.__vtbl_pt
 000013b0: 725f 7479 7065 0069 6e74 0062 6f6f 6c61  r_type.int.boola
 000013c0: 6c70 6861 0066 6d74 666c 6167 7300 756e  lpha.fmtflags.un
 000013d0: 7369 676e 6564 2069 6e74 0064 6563 0066  signed int.dec.f
 000013e0: 6978 6564 0068 6578 0069 6e74 6572 6e61  ixed.hex.interna
@@ -5682,15 +5682,15 @@
 00016310: 2f69 6e63 6c75 6465 2f6d 616c 6c6f 6300  /include/malloc.
 00016320: 2f75 7372 2f69 6e63 6c75 6465 2f73 7973  /usr/include/sys
 00016330: 002f 5573 6572 732f 6361 726c 6f73 2f44  ./Users/carlos/D
 00016340: 6f63 756d 656e 7473 2f64 6576 656c 6f70  ocuments/develop
 00016350: 6572 2f72 6570 726f 6475 6369 626c 652d  er/reproducible-
 00016360: 6275 696c 6473 2f73 616e 6462 6f78 2f68  builds/sandbox/h
```

Microsoft Visual Studio: sin opciones, parcheo directo

gcc: flags del compilador

- -fdebug-prefix-map=OLD=NEW can strip directory prefixes from debug info.
- -fmacro-prefix-map=OLD=NEW (desde 8.0)
- -ffile-prefix-map=OLD=NEW (desde 8.0)

clang: -fdebug-prefix-map=OLD=NEW (clang>3.8)

```
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
a87d15bf73dc80a6ccf66ea448e65d70  srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
a87d15bf73dc80a6ccf66ea448e65d70  srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
1f1765dde2b67f057b34fb86cd693945  srcA/build/libHelloLib.a
1f1765dde2b67f057b34fb86cd693945  srcB/build/libHelloLib.a
6d977adfac0e22c46446637d17e74db7  srcA/build/hello
6d977adfac0e22c46446637d17e74db7  srcB/build/hello
```

```
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ a87d15bf73dc80a6ccf66ea448e65d70  srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
a87d15bf73dc80a6ccf66ea448e65d70  srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
1f1765dde2b67f057b34fb86cd693945  srcA/build/libHelloLib.a
1f1765dde2b67f057b34fb86cd693945  srcB/build/libHelloLib.a
6d977adfac0e22c46446637d17e74db7  srcA/build/hello
6d977adfac0e22c46446637d17e74db7  srcB/build/hello
```

```
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ a87d15bf73dc80a6ccf66ea448e65d70  srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
a87d15bf73dc80a6ccf66ea448e65d70  srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
✅ 1f1765dde2b67f057b34fb86cd693945  srcA/build/libHelloLib.a
1f1765dde2b67f057b34fb86cd693945  srcB/build/libHelloLib.a
6d977adfac0e22c46446637d17e74db7  srcA/build/hello
6d977adfac0e22c46446637d17e74db7  srcB/build/hello
```

```
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
[ 25%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 50%] Linking CXX static library libHelloLib.a
[ 50%] Built target HelloLib
[ 75%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ a87d15bf73dc80a6ccf66ea448e65d70  srcA/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
a87d15bf73dc80a6ccf66ea448e65d70  srcB/build/CMakeFiles/HelloLib.dir/hello_world.cpp.o
✅ 1f1765dde2b67f057b34fb86cd693945  srcA/build/libHelloLib.a
1f1765dde2b67f057b34fb86cd693945  srcB/build/libHelloLib.a
✅ 6d977adfac0e22c46446637d17e74db7  srcA/build/hello
6d977adfac0e22c46446637d17e74db7  srcB/build/hello
```

Orden de los ficheros


.
├── CMakeLists.txt
├── hello_world.cpp
├── hello_world.hpp
├── main.cpp
├── sources0.cpp
├── sources0.hpp
├── sources1.cpp
├── sources1.hpp
├── sources2.cpp
└── sources2.hpp

```
cmake_minimum_required(VERSION 3.0)
project(HelloWorld)
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED ON)
add_library(HelloLib hello_world.cpp
                     sources0.cpp
                     sources1.cpp
                     sources2.cpp)
add_executable(hello main.cpp)
target_link_libraries(hello HelloLib)
```

```
cmake_minimum_required(VERSION 3.0)
project(HelloWorld)
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED ON)
add_library(HelloLib hello_world.cpp
                     sources1.cpp
                     sources0.cpp
                     sources2.cpp)
add_executable(hello main.cpp)
target_link_libraries(hello HelloLib)
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
285b351de49febad923b00165dac1618  sources0A.cpp.o
285b351de49febad923b00165dac1618  sources0B.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
e2f5d0fc5366dadf6c72c68dec09097b  helloA
74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
285b351de49febad923b00165dac1618  sources0A.cpp.o
285b351de49febad923b00165dac1618  sources0B.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
e2f5d0fc5366dadf6c72c68dec09097b  helloA
74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0A.cpp.o
285b351de49febad923b00165dac1618  sources0B.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
e2f5d0fc5366dadf6c72c68dec09097b  helloA
74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0A.cpp.o
285b351de49febad923b00165dac1618  sources0B.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
e2f5d0fc5366dadf6c72c68dec09097b  helloA
74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
   c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0A.cpp.o
   285b351de49febad923b00165dac1618  sources0B.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
   36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
✅ 5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
   5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
   03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
   0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
   e2f5d0fc5366dadf6c72c68dec09097b  helloA
   74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0A.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0B.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
✅ 5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
✅ 5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
❌ 03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
❌ 0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
❌ e2f5d0fc5366dadf6c72c68dec09097b  helloA
❌ 74bbb9471ec9ed96ab986d4128c43531  helloB
```

```
[100%] Linking CXX executable hello
[100%] Built target hello
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/carlos/Documents/developer…
[ 14%] Building CXX object CMakeFiles/HelloLib.dir/hello_world.cpp.o
[ 28%] Building CXX object CMakeFiles/HelloLib.dir/sources0.cpp.o
[ 42%] Building CXX object CMakeFiles/HelloLib.dir/sources1.cpp.o
[ 57%] Building CXX object CMakeFiles/HelloLib.dir/sources2.cpp.o
[ 71%] Linking CXX static library libHelloLib.a
[ 71%] Built target HelloLib
[ 85%] Building CXX object CMakeFiles/hello.dir/main.cpp.o
[100%] Linking CXX executable hello
[100%] Built target hello
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldA.cpp.o
✅ c04a58134ff17df6bf5bcb672306135c  hello_worldB.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0A.cpp.o
✅ 285b351de49febad923b00165dac1618  sources0B.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1A.cpp.o
✅ 36ffb5fbb0b60ed32b1b9f5d2a93e3c4  sources1B.cpp.o
✅ 5d353dcd4b6d566c1527d21701c71eda  sources2A.cpp.o
✅ 5d353dcd4b6d566c1527d21701c71eda  sources2B.cpp.o
❌ 03fb031d1d49da672b3c42c317fcabd7  libHelloLibA.a
❌ 0c67130e22f3988102da8c50c5b6a56a  libHelloLibB.a
❌ e2f5d0fc5366dadf6c72c68dec09097b  helloA
❌ 74bbb9471ec9ed96ab986d4128c43531  helloB
```

Aleatoriedad

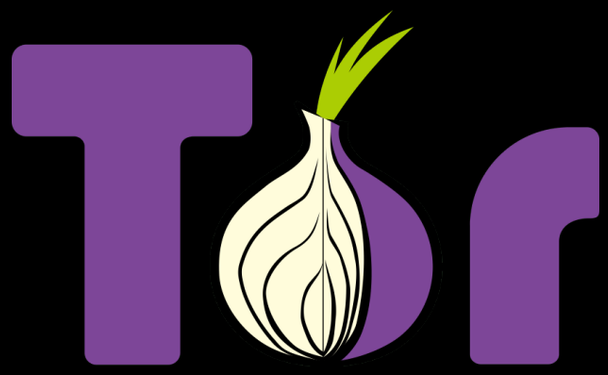
gcc: Link-Time Optimizations (-flto flag)

- Mete nombres generados de forma aleatoria en los binarios
- Usar -frandom-seed flag para dar una semilla que gcc usa en vez de generar los numeres aleatorios. Esta opción es por fichero

```
set(LIB_SOURCES
    ./src/source1.cpp
    ./src/source2.cpp
    ./src/source3.cpp)

foreach(_file ${LIB_SOURCES})
    file(SHA1 ${_file} checksum)
    string(SUBSTRING ${checksum} 0 8 checksum)
    set_property(SOURCE ${_file} APPEND_STRING
PROPERTY COMPILE_FLAGS "-frandom-seed=0x${
checksum}")
endforeach()
```

Más información



Reproducible Builds

yocto PROJECT

An introduction to deterministic builds with C/C++

Sep 2, 2019 • 24 minutes to read

What are deterministic builds?

A deterministic build is a process of building the same source code with the same build environment and build instructions producing the same binary in two builds, even if they are made on different machines, build directories and with different names. They are also sometimes called reproducible or hermetic builds if it is guaranteed to produce the same binaries even compiling from different folders.

Deterministic builds are not something that happens naturally. Normal projects do not produce deterministic builds and the reasons that they are not produced can be different for each operating system and compiler.

Deterministic builds should be guaranteed for a given *build environment*. That means that certain variables such as the *operating system*, *build system versions* and *target architecture* are assumed to remain the same between different builds.

There are lots of efforts coming from different organizations in the past years to achieve deterministic builds such as Chromium, Reproducible builds, or Yocto.

Gracias!