GCC Compilation Process

Device used – M1 MacBook Air

Submitted by – Anushthan Saxena (S20210010027)

Starting with this simple hello program

```
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↪ ls
hello.c
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↪ cat hello.c
#include <stdio.h>
int main(){
    printf("Hello");
    return 0;
}
```

- ***Preprocessing:*** We create a ".i" file using " **cpp hello.c > hello.i** ". However, this program gives multiple errors regarding system architecture difference in M1 macs.

```
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:49: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:66: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                                 ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:83: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                                                  ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:100: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                                                                    ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:117: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                                                                                        ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4488:134: error: _Pragma takes a parenthesized string literal
    #define __API_AVAILABLE_BEGIN6(a,b,c,d,e,f) __API_A_BEGIN(a) __API_A_BEGIN(b) __API_A_BEGIN(c) __API_A_BEGIN(d) __API_A_BEGIN(e) __API_A_BEGIN(f)
                                                                                                                                    ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4481:30: note: expanded from macro '__API_A_BEGIN'
    #define __API_A_BEGIN(x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_AVAILABLE_PLATFORM_##x))), apply_to = __API_APPLY_TO)))
                             ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4528:50: error: _Pragma takes a parenthesized string literal
    #define __API_DEPRECATED_BEGIN_MSG3(msg,a,b) __API_D_BEGIN(msg,a) __API_D_BEGIN(msg,b)
                                                 ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h:4525:35: note: expanded from macro '__API_D_BEGIN'
    #define __API_D_BEGIN(msg, x) _Pragma(__API_RANGE_STRINGIFY (clang attribute (__attribute__((availability(__API_DEPRECATED_PLATFORM_##x,message=msg))), apply_to = __API_APPLY_TO)))
                                  ^
fatal error: too many errors emitted, stopping now [-ferror-limit=]
20 errors generated.
```

So we use the alternative **“gcc -E -o hello.i hello.c”**

```
[↪ gcc -E hello.c -o hello.i
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ ls
 hello.c  hello.i
```

- ***Compilation:*** next we convert this “.i” file to a “.s” file using **“gcc -S hello.i”**

```
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ gcc -S hello.i
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ ls
 hello.c  hello.i  hello.s
```

- ***Assembly:*** The assembler will convert this “.s” code to an object file “.o”, using the command “**as -o hello.o hello.s”**

```
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ ls
 hello.c  hello.i  hello.s
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ as hello.s -o hello.o
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ ls
 hello.c  hello.i  hello.o  hello.s
```

- ***Linker:*** Normaly, the linker links the object code with the library code to produce an executable file with the command “**ld -o hello.out hello.o <library_path> -lc”**

  This library path can be found out using “**which printf”.**

```
 anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
[↪ which printf
 /usr/bin/printf
```

Here we run into a problem regarding M1 Architecture. Linker doesn't work by itself for the aarch64 ARM architecture which M1 has.

```
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↳ uname -a
Darwin Anushthans-MacBook-Air.local 21.4.0 Darwin Kernel Version 21.4.0: Fri Mar 18 00:47:26 PDT 20
22; root:xnu-8020.101.4~15/RELEASE_ARM64_T8101 arm64
```

It is shown with an example below:

```
anushthan at Anushthans-MacBook-Air in ~/D/test
↳ make btest
gcc -O -Wall -m32 -lm -o btest bits.c btest.c decl.c tests.c
ld: unknown/unsupported architecture name for: -arch armv4t
clang: error: linker command failed with exit code 1 (use -v to see invocation)
make: *** [btest] Error 1
```

```
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↳ ld -o hello.out hello.o /usr/bin/printf
ld: can't link with a main executable file '/usr/bin/printf' for architecture arm64
```

Therefore, we directly use the Apple gcc command to get the final executable file.

```
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↳ gcc -o hello.o hello.c
anushthan at Anushthans-MacBook-Air in ~/D/C/L/Assembler
↳ ./hello.o
Hello
```