

C++ Montréal



Gabriel
Aubut-Lussier



Druid^e

Write, Build and Execute...

...is is really that simple?

2017-05-30



Write, Build and Execute...

...is it really that simple?

2017-05-30

To IDE or Not to IDE?

Does relying on an IDE prevent you from learning ?

Write

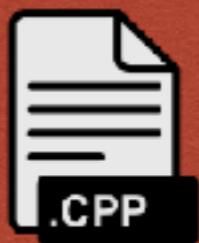
Build

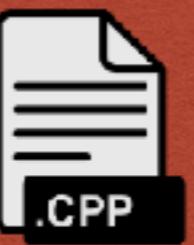
Execute

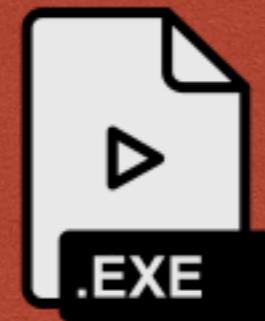
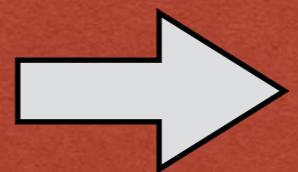
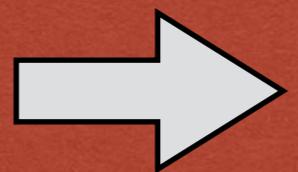
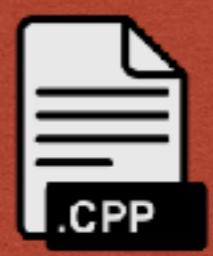
Build

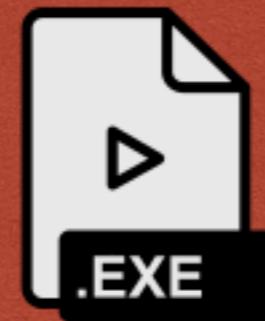
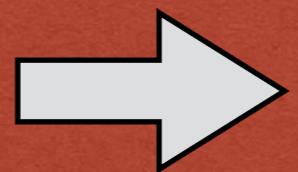
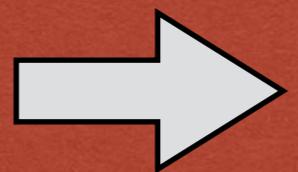
Execute

Build

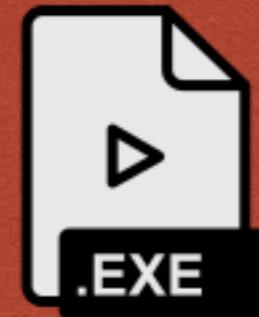
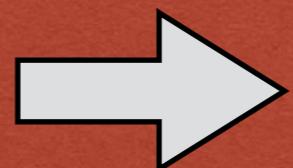
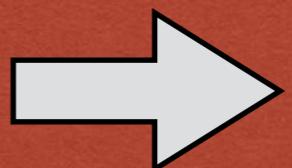


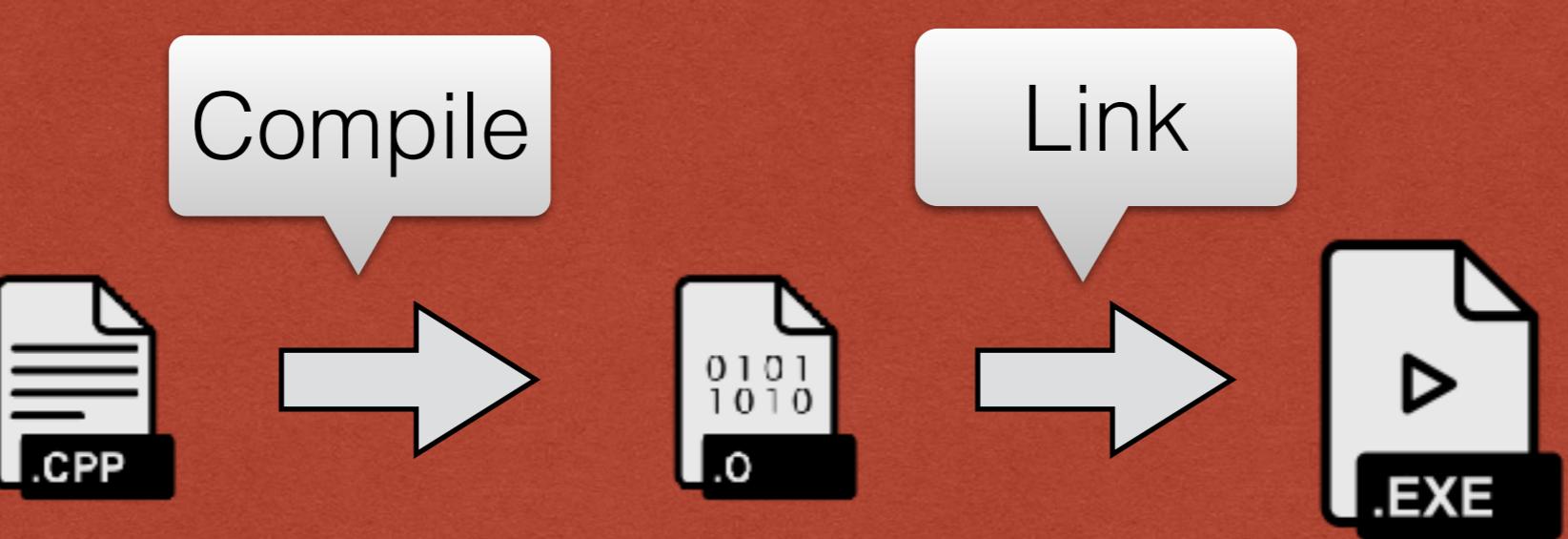


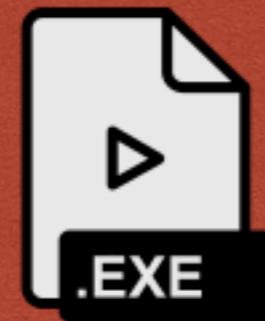
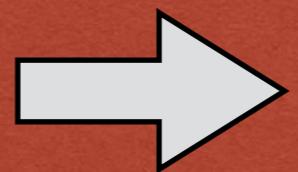
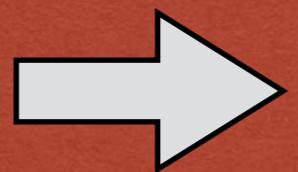
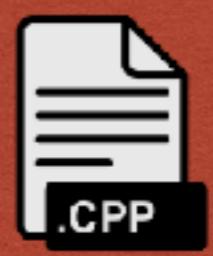


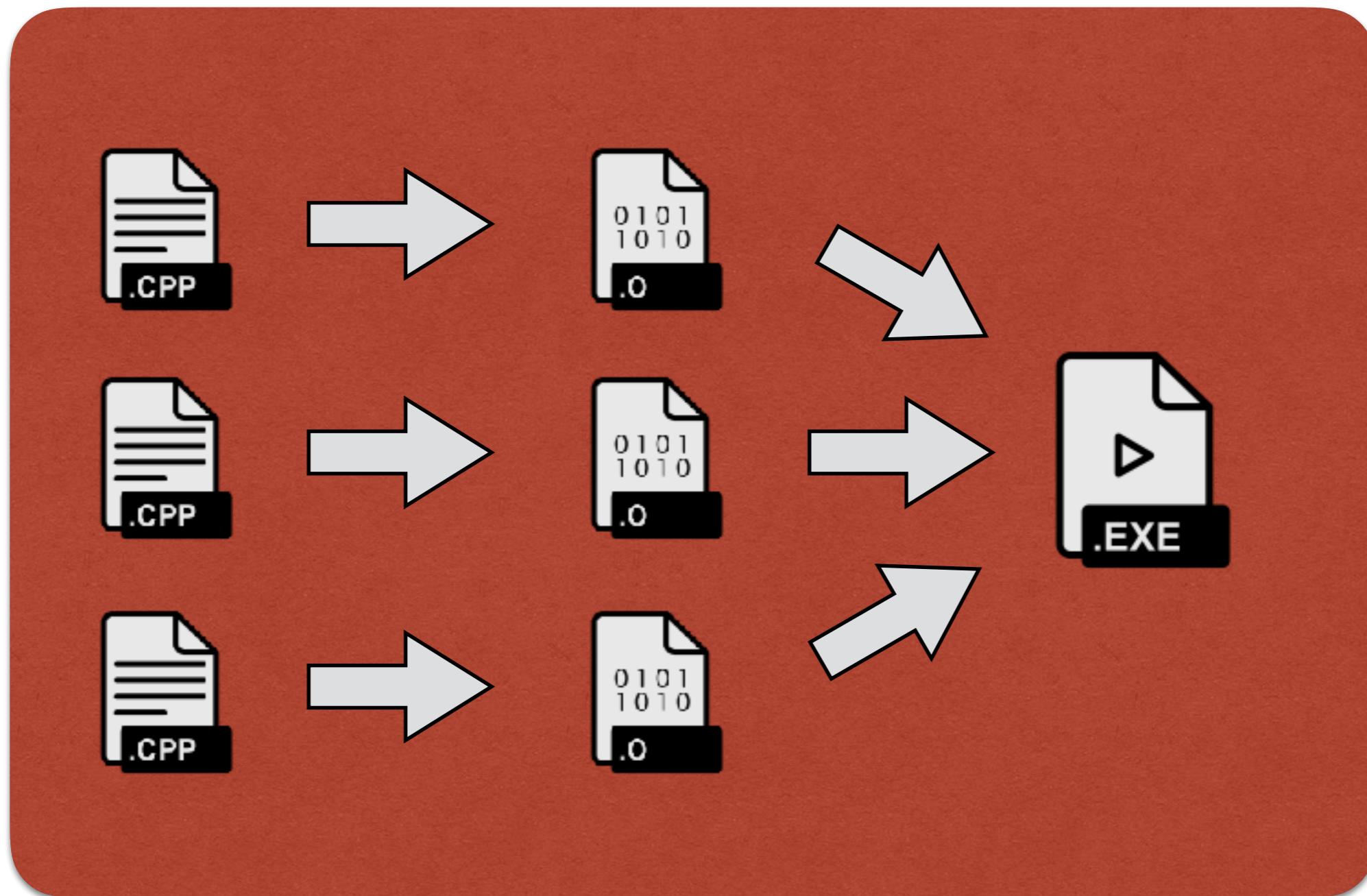


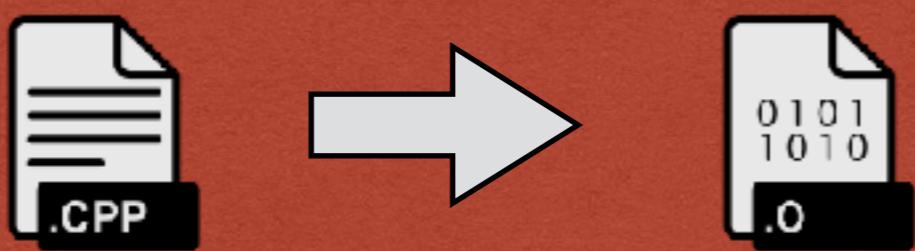
Compile

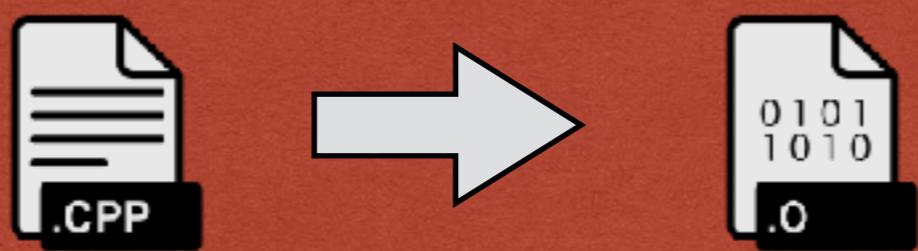


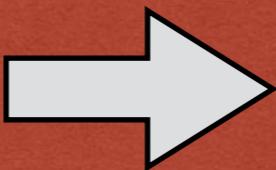
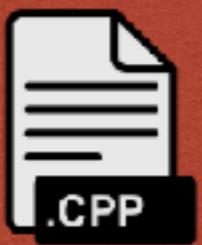


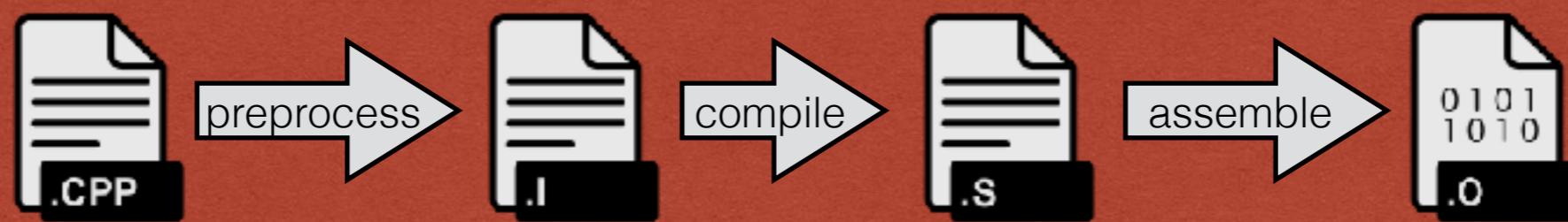


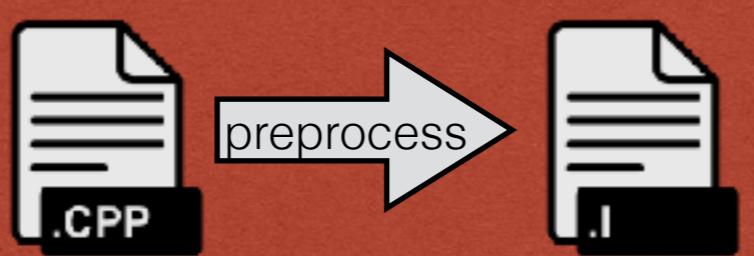




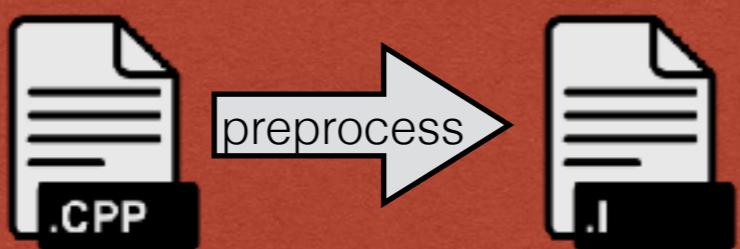




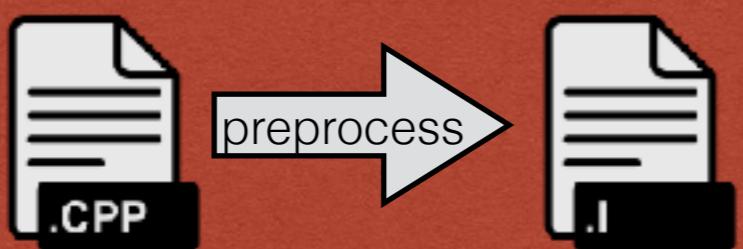




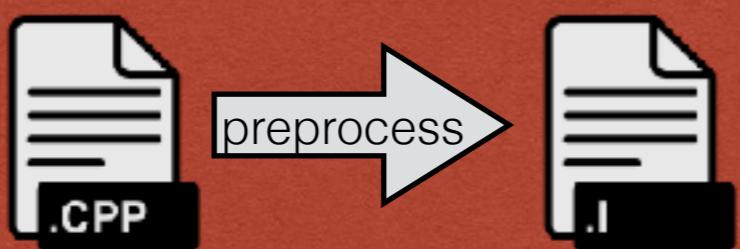
```
clang++ -E main.cpp -o main.i
```

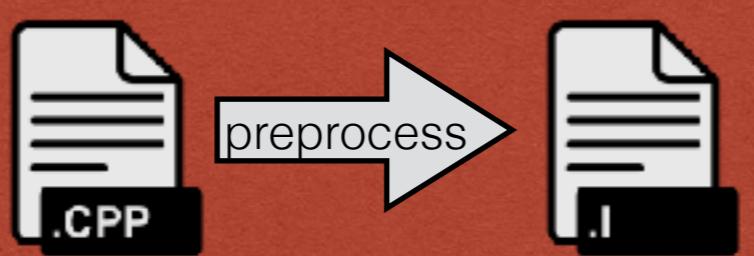


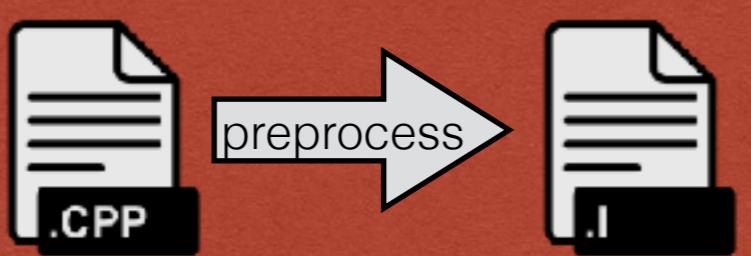
```
clang++ -E main.cpp -o main.i
```



```
clang++ -E main.cpp -o main.i
```







main.cpp

```
1 #include "CloudBasedSuperAI.h"
```

```
2
```

```
3 int main()
```

```
4 {
```

```
5     runCloudBasedSuperAI();
```

```
6     return 0;
```

```
7 }
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15
```

```
16
```

```
17
```

```
18
```

```
19
```

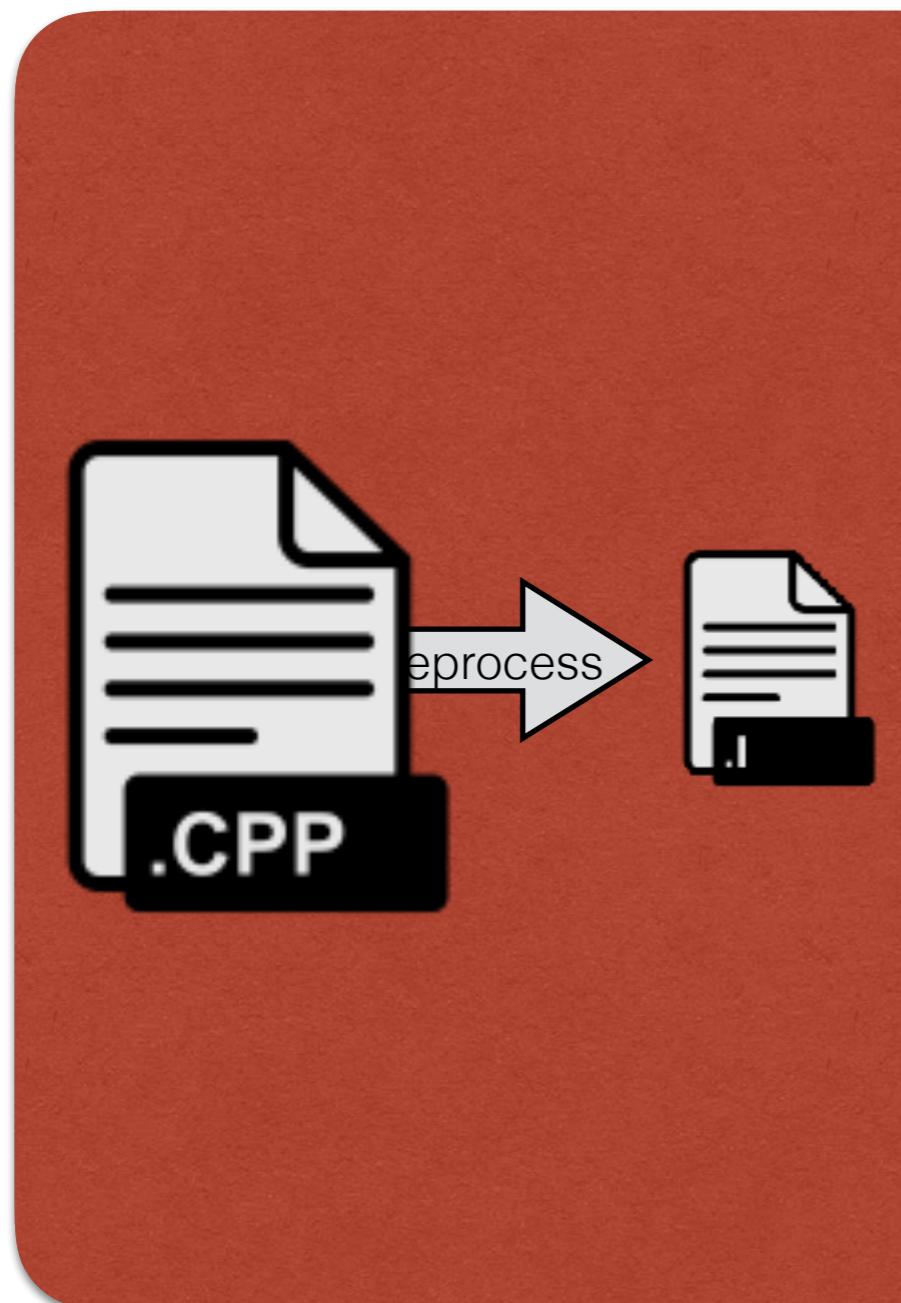
```
20
```

```
21
```

```
22
```

```
23
```

```
24
```



main.cpp

```
1 #include "CloudBasedSuperAI.h"
```

```
2
```

```
3 int main()
```

```
4 {
```

```
5     runCloudBasedSuperAI();
```

```
6     return 0;
```

```
7 }
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15
```

```
16
```

```
17
```

```
18
```

```
19
```

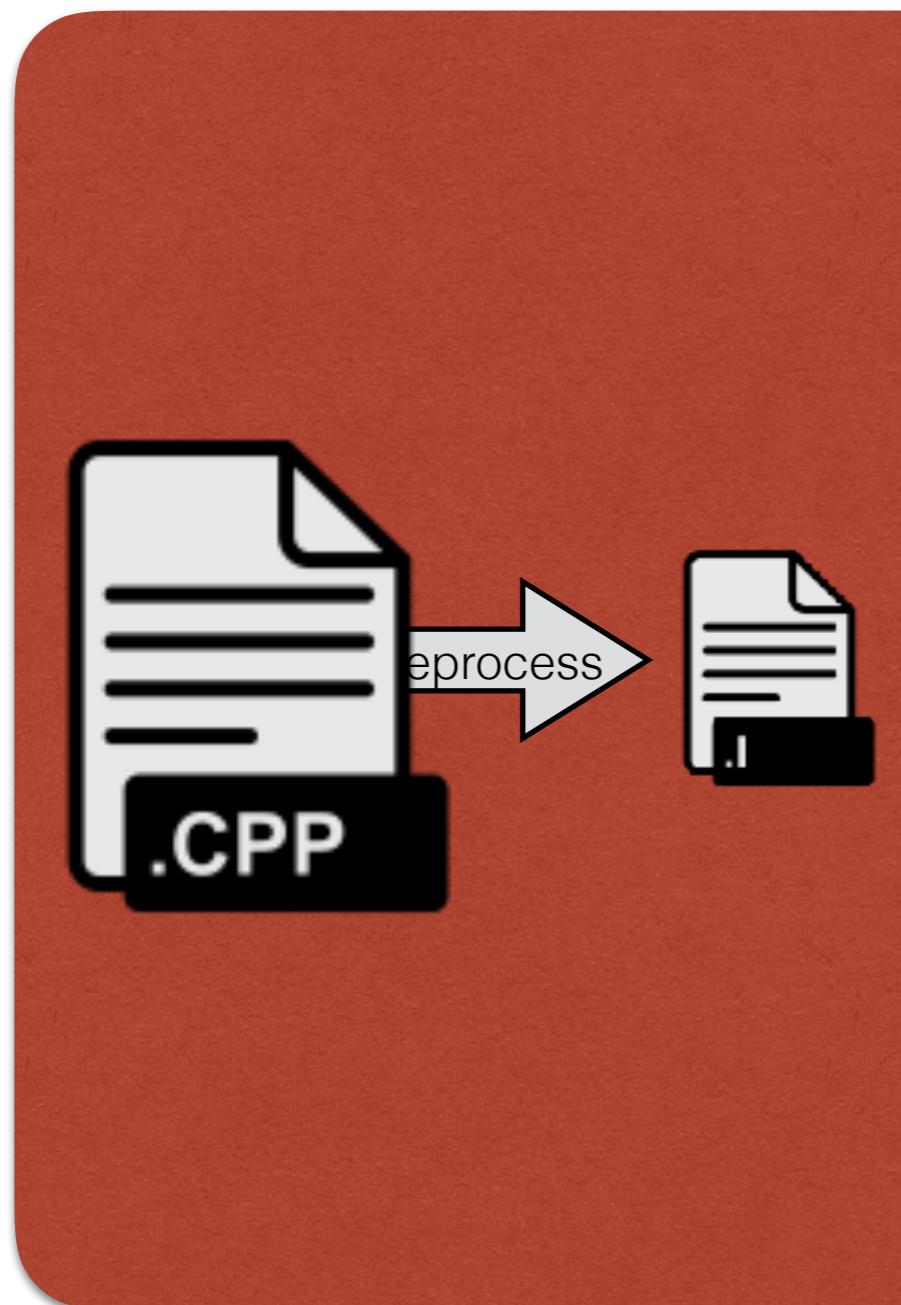
```
20
```

```
21
```

```
22
```

```
23
```

```
24
```



main.cpp

```
1 #include "CloudBasedSuperAI.h"
```

```
2
```

```
3 int main()
```

```
4 {
```

```
5     runCloudBasedSuperAI();
```

```
6     return 0;
```

```
7 }
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15
```

```
16
```

```
17
```

```
18
```

```
19
```

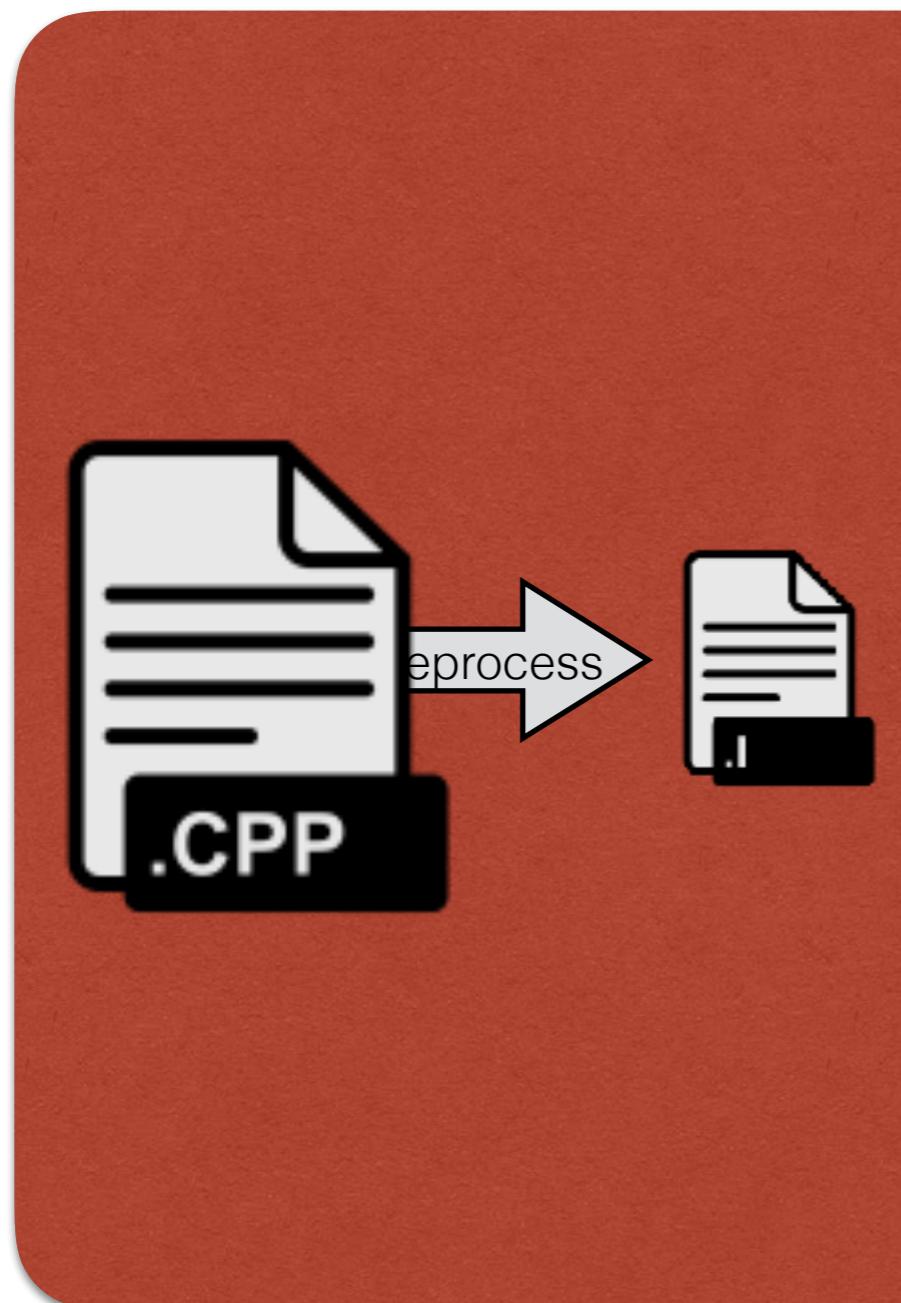
```
20
```

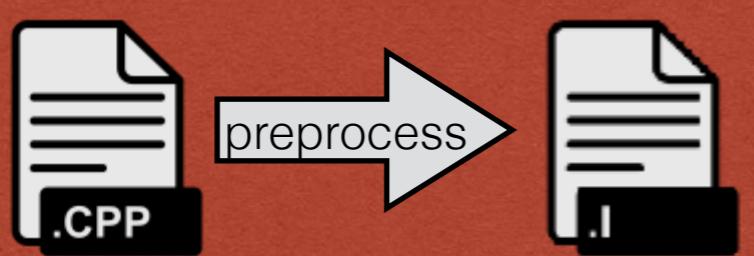
```
21
```

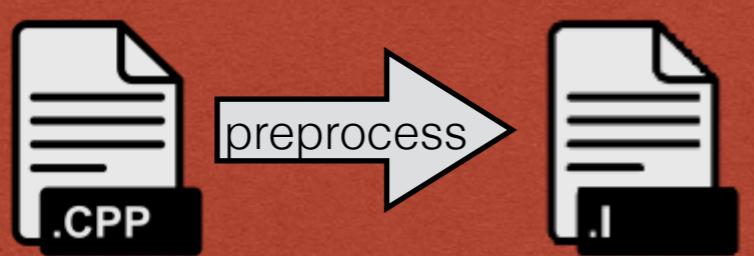
```
22
```

```
23
```

```
24
```









#include

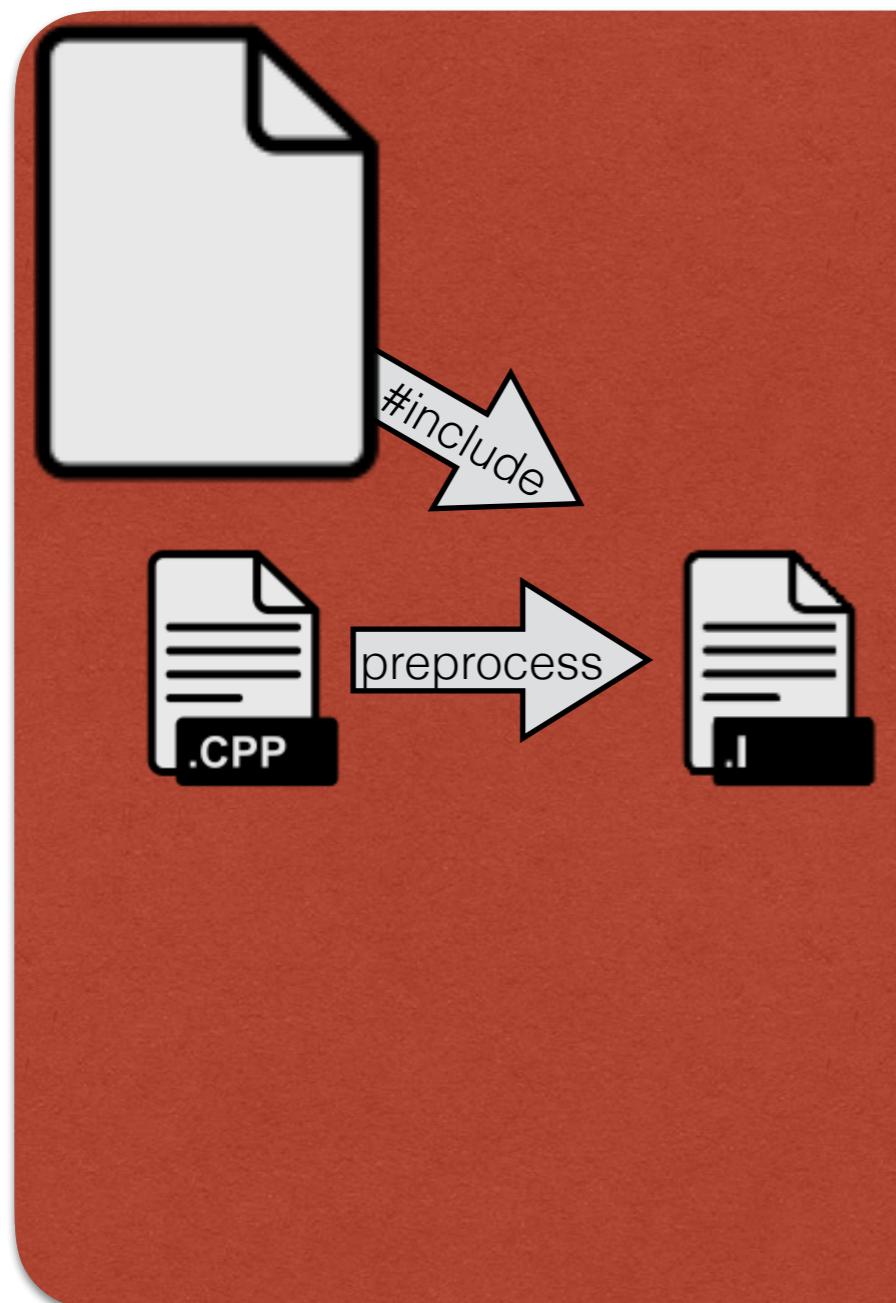
A white arrow pointing from the header file icon towards the CPP file icon, labeled with the text "#include".

preprocess

A white arrow pointing from the CPP file icon towards the I file icon, labeled with the text "preprocess".

CloudBasedSuperAI.h

```
1 #ifndef CloudBasedSuperAI_h  
2 #define CloudBasedSuperAI_h  
3  
4 void runCloudBasedSuperAI();  
5  
6 #endif /* CloudBasedSuperAI_h */  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24
```

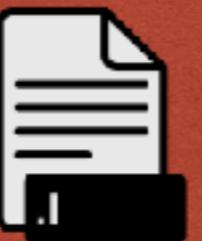




#include

A white arrow pointing from the header file icon towards the CPP file icon, labeled with the text "#include".

preprocess

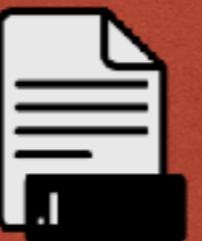
A white arrow pointing from the CPP file icon towards the I file icon, labeled with the text "preprocess".



#include

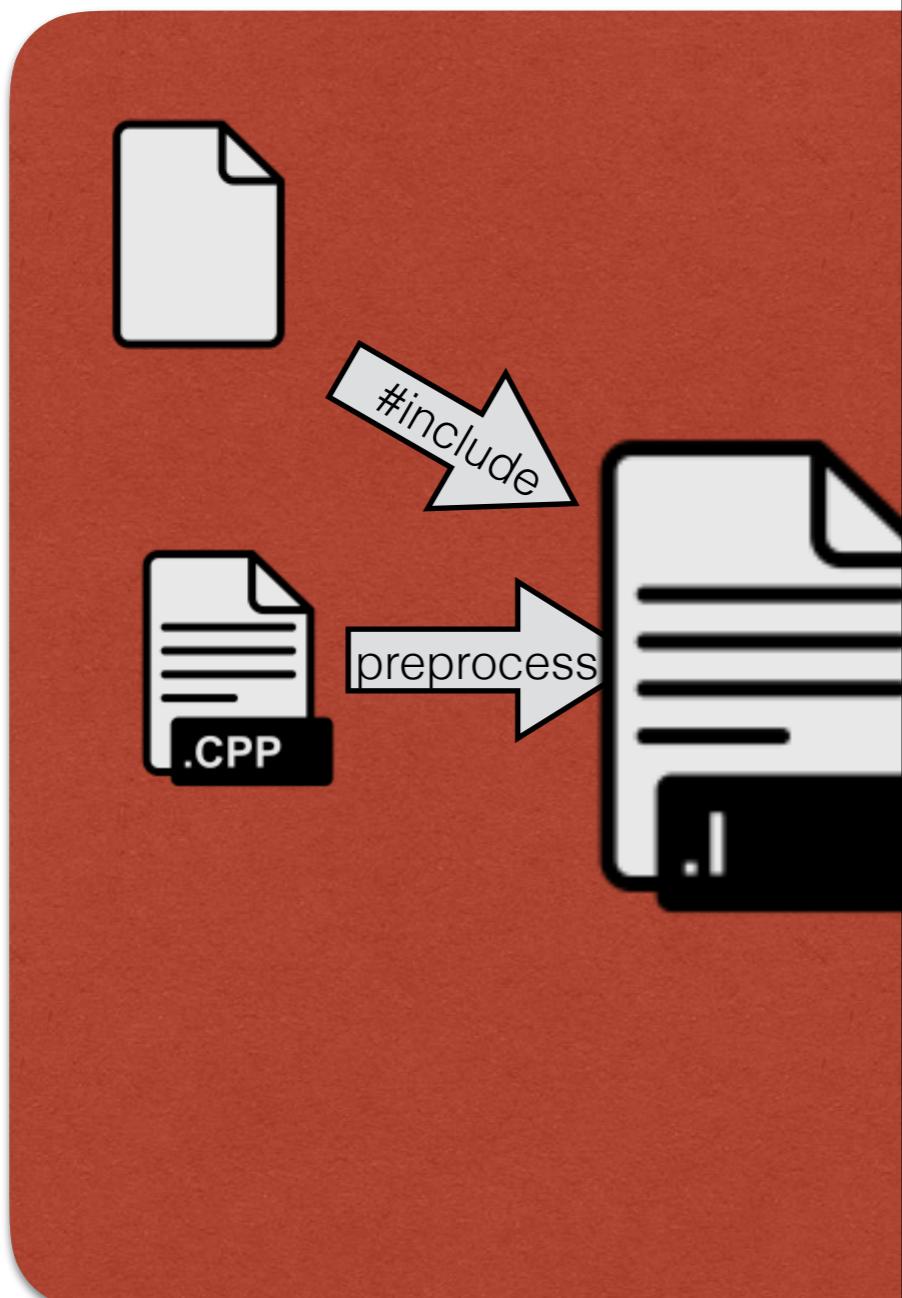
A white arrow pointing from the header file icon towards the CPP file icon, labeled with the text "#include".

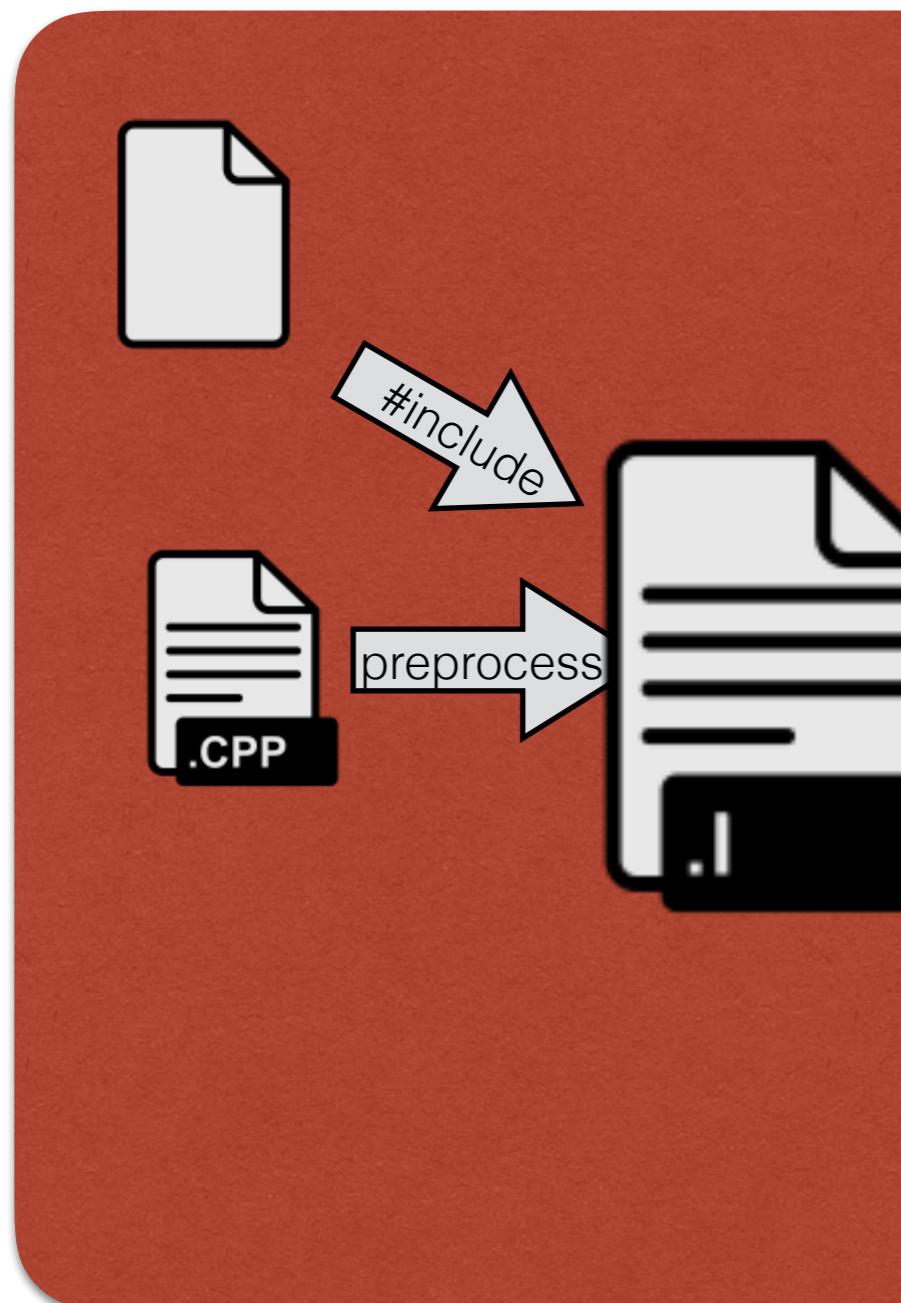
preprocess

A white arrow pointing from the CPP file icon towards the I file icon, labeled with the text "preprocess".

main.i

```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17     runCloudBasedSuperAI();
18     return 0;
19 }
20
21
22
23
24
```



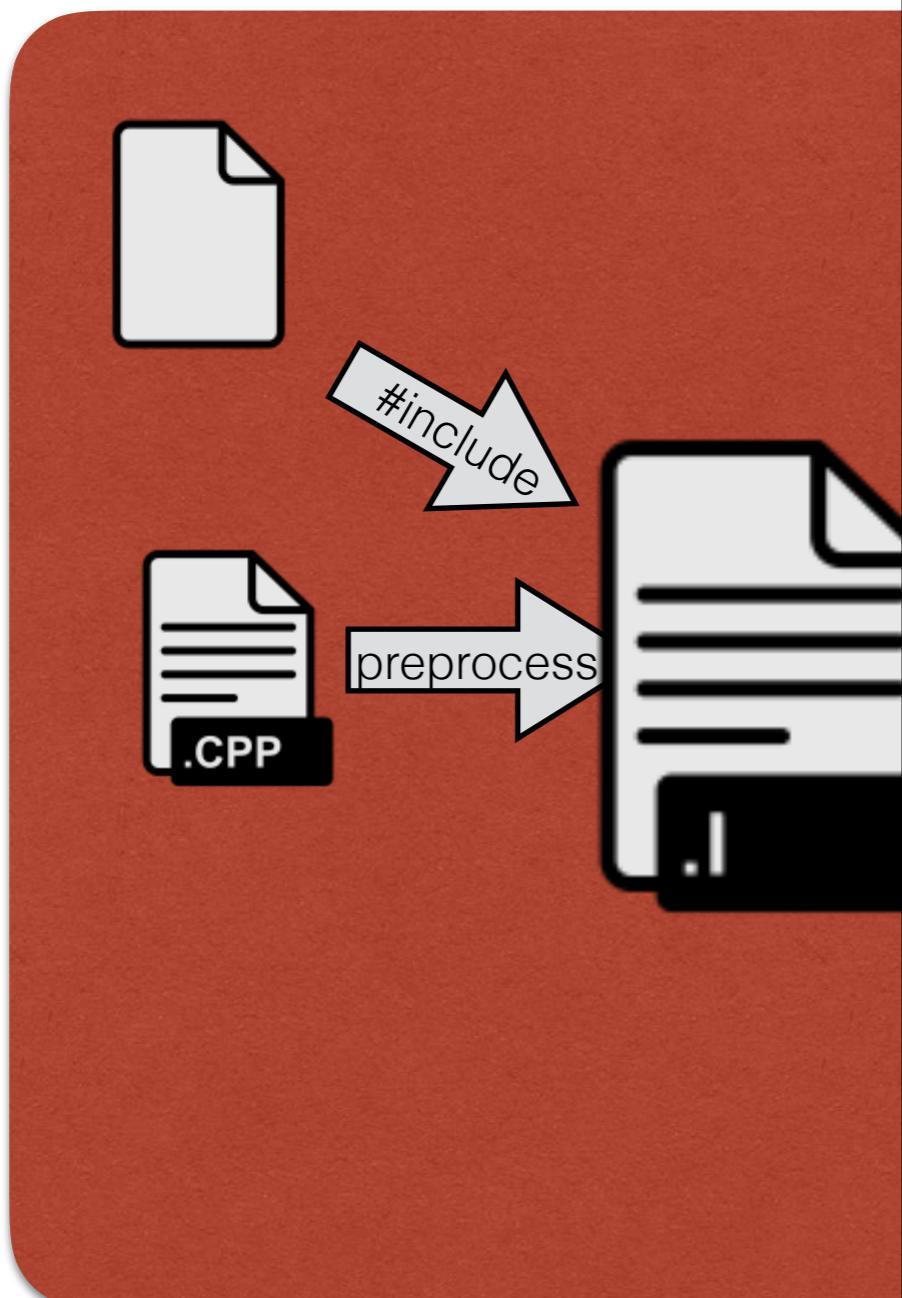


main.i

```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17     runCloudBasedSuperAI();
18     return 0;
19 }
20
21
22
23
24
```

main.i

```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17     runCloudBasedSuperAI();
18     return 0;
19 }
20
21
22
23
24
```

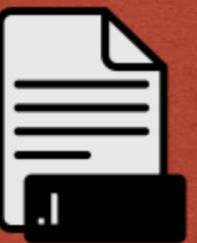


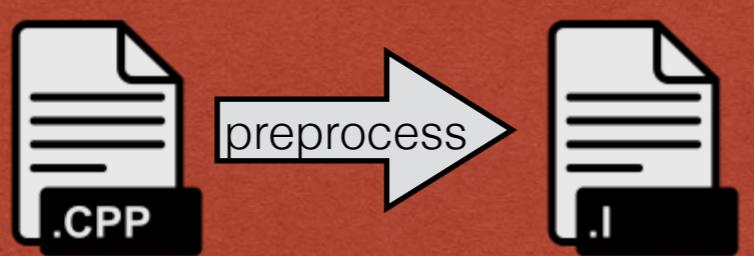


#include

A white arrow pointing from the header file icon towards the CPP file icon, labeled with the text "#include".

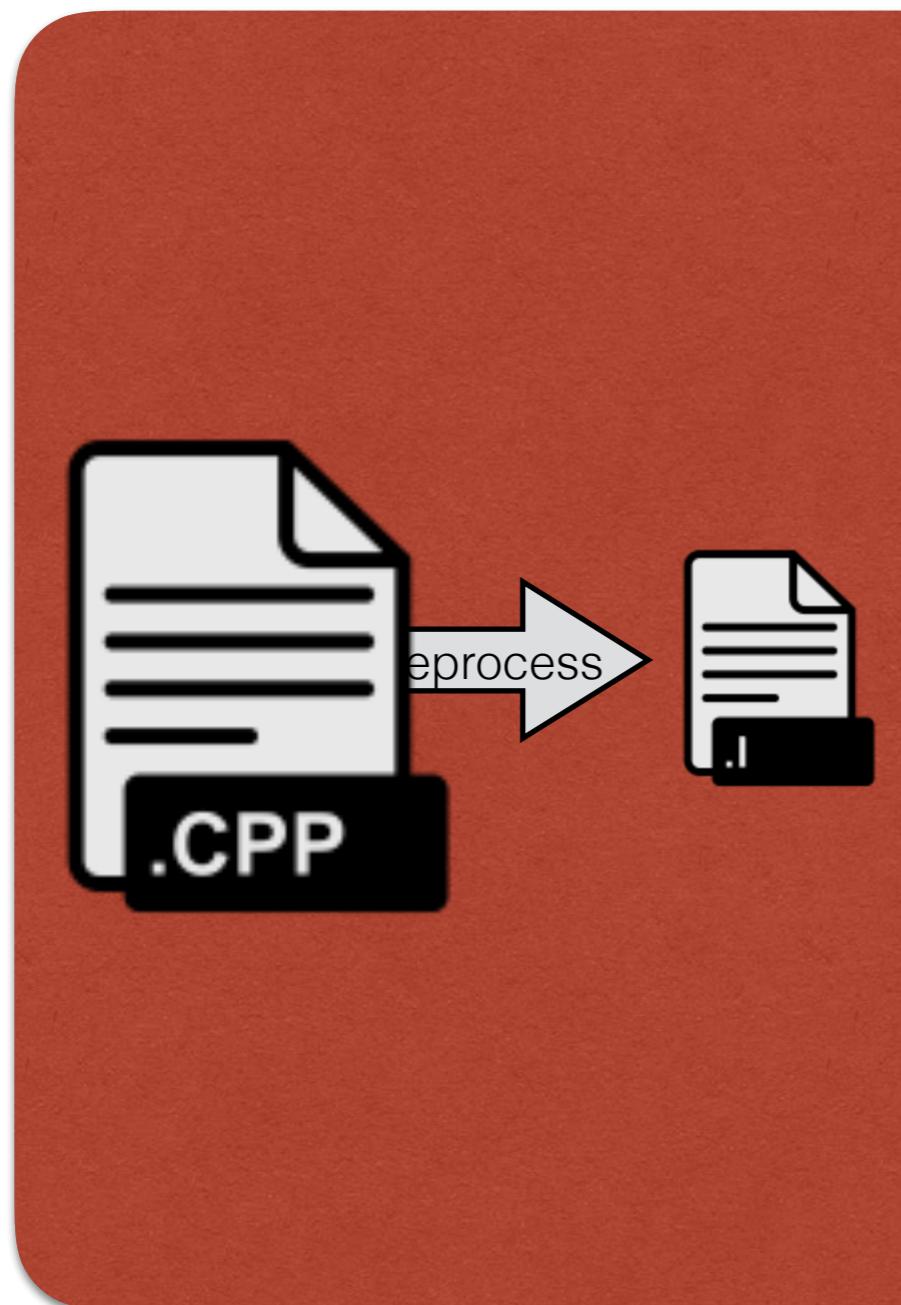
preprocess

A white arrow pointing from the CPP file icon towards the I file icon, labeled with the text "preprocess".



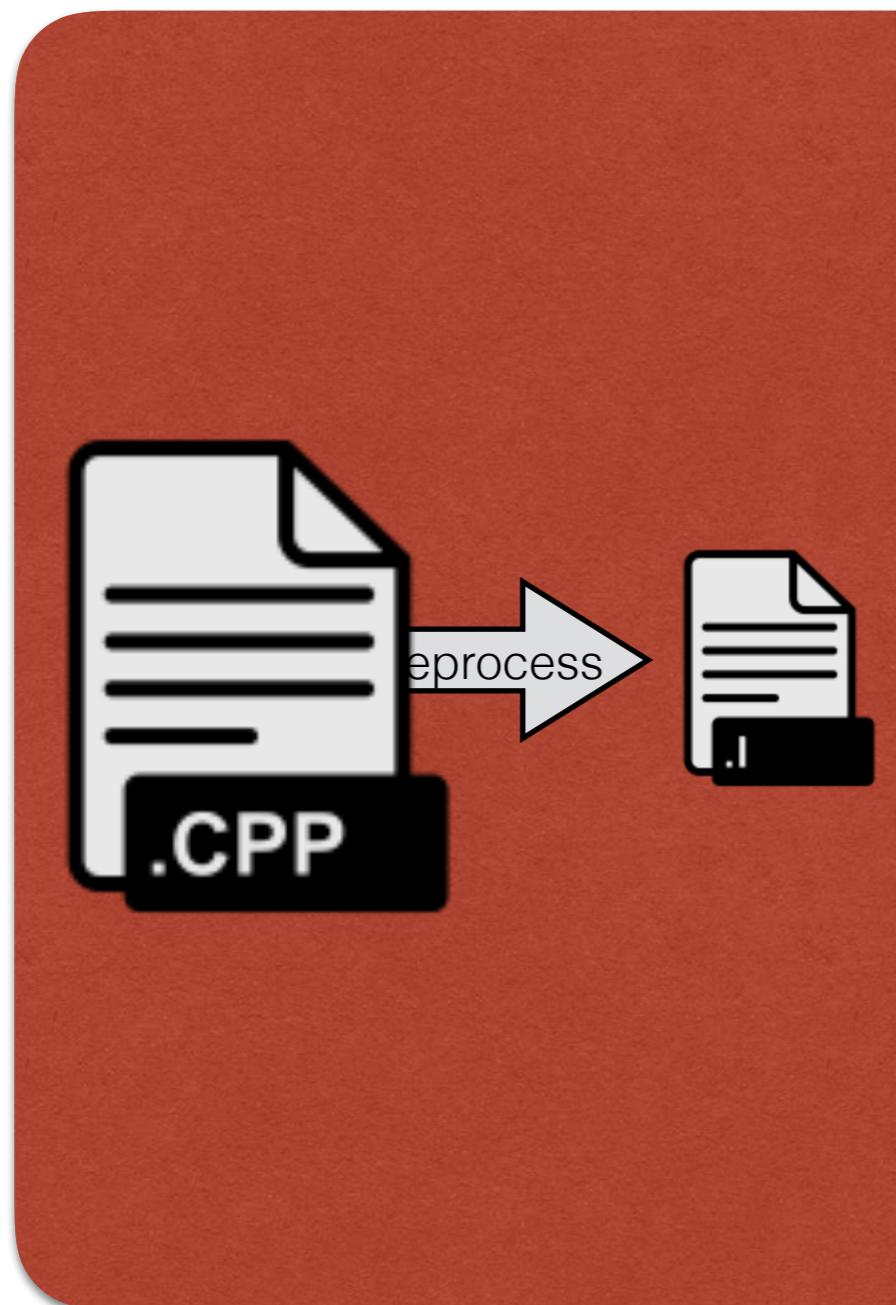
main.cpp

```
1 #include <vector>
2 #include "CloudBasedSuperAI.h"
```



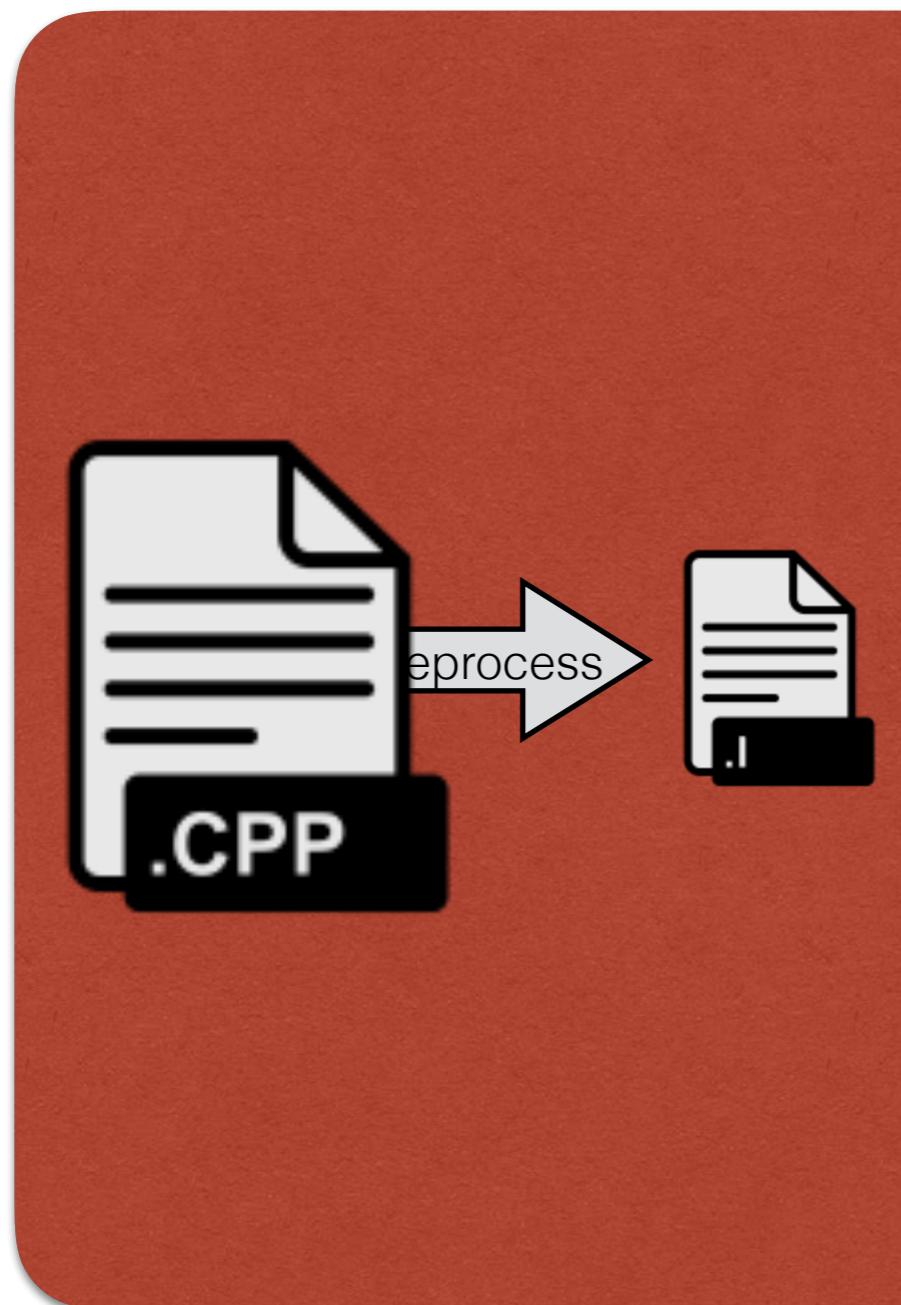
main.cpp

```
1 #include <vector>
2 #include "CloudBasedSuperAI.h"
```



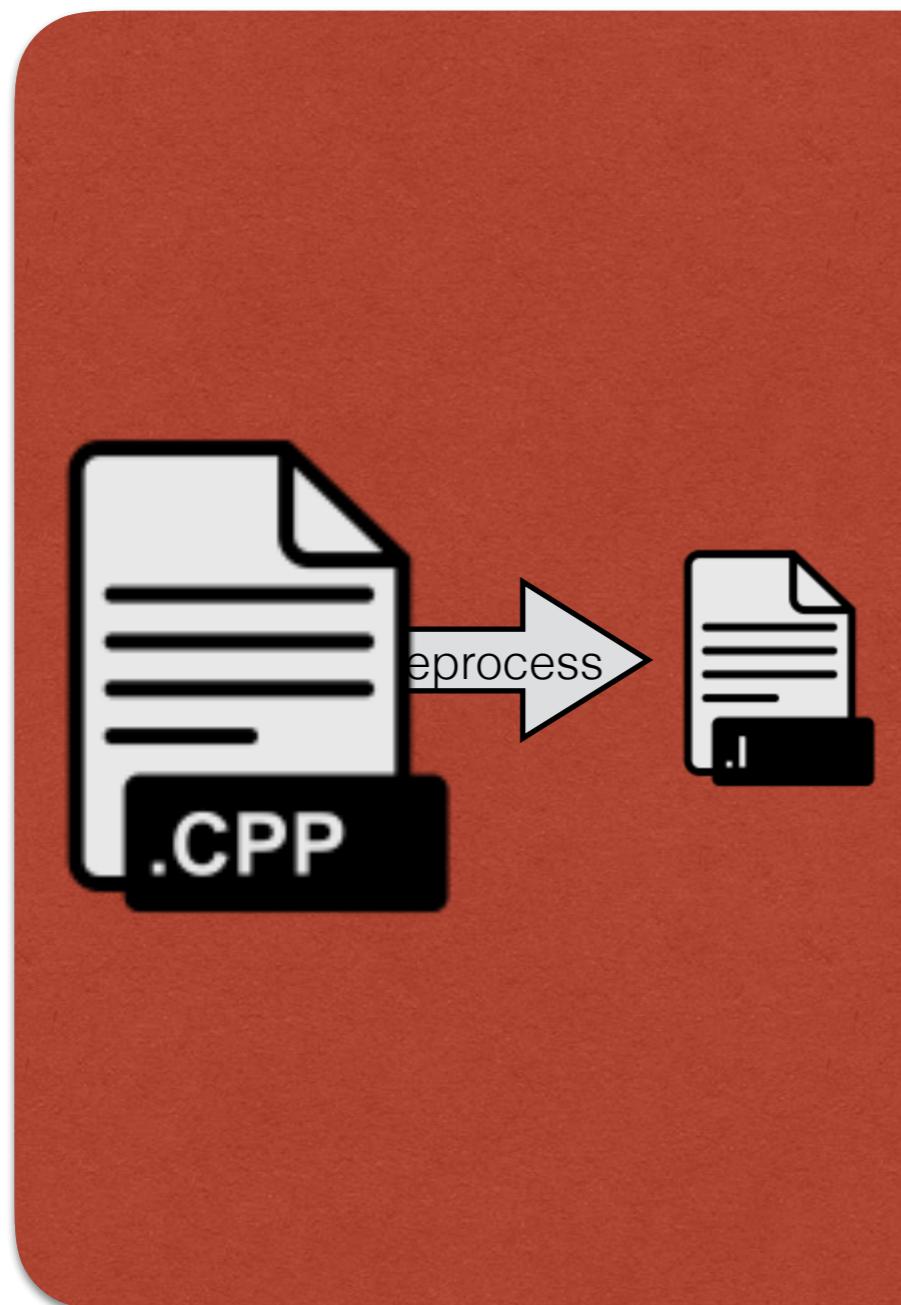
main.cpp

```
1 #include <vector>
2 #include "CloudBasedSuperAI.h"
```



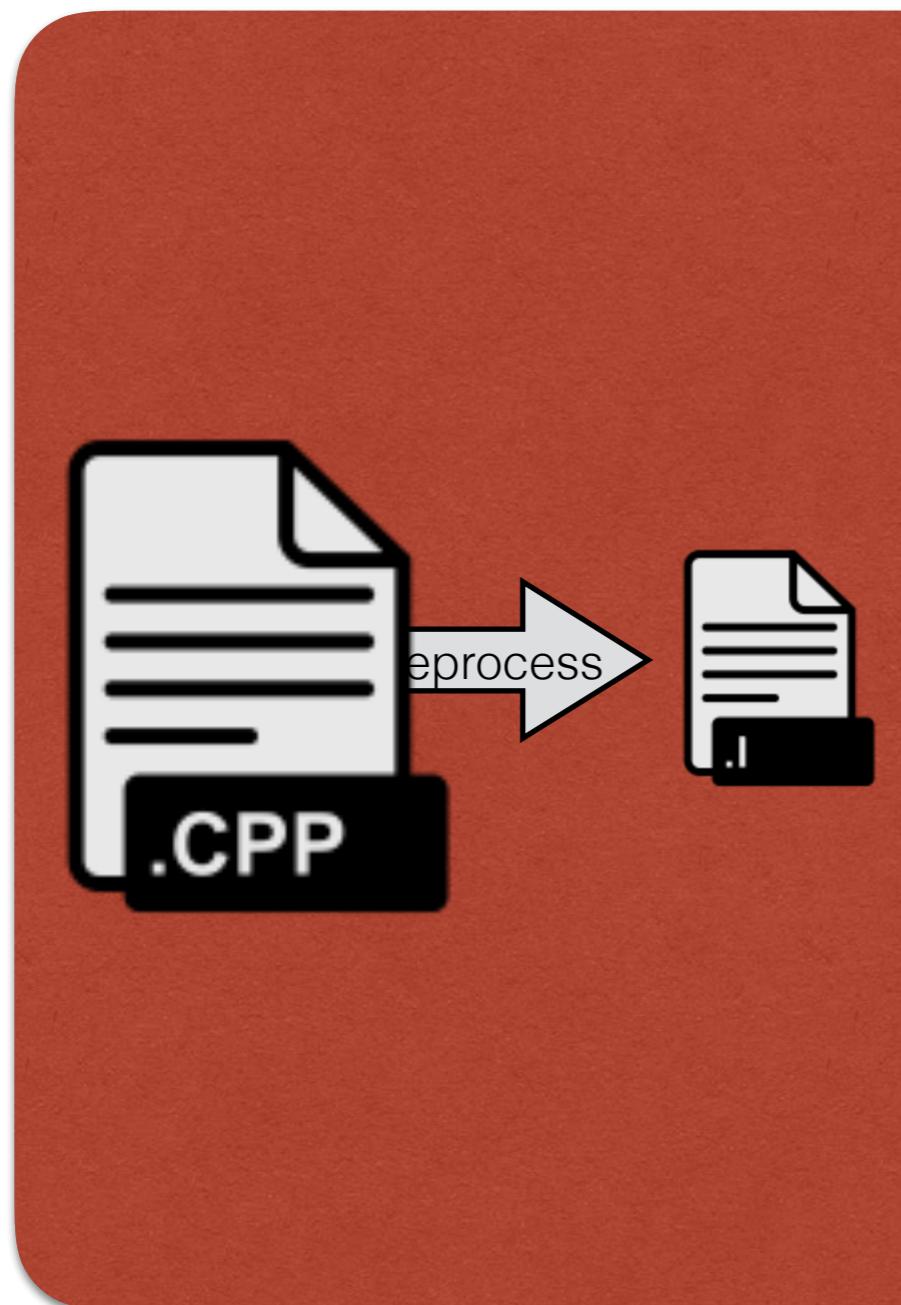
main.cpp

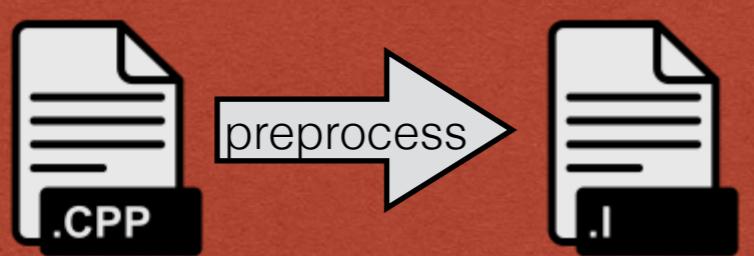
```
1 #include <vector>
2 #include "CloudBasedSuperAI.h"
```

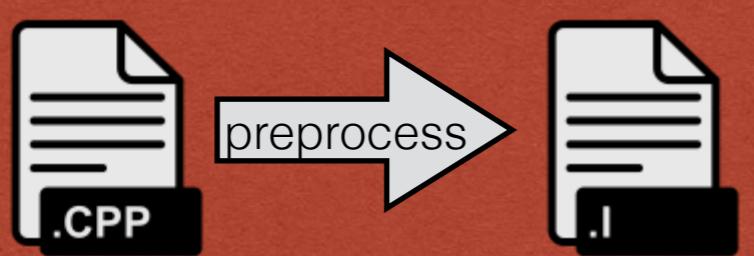


main.cpp

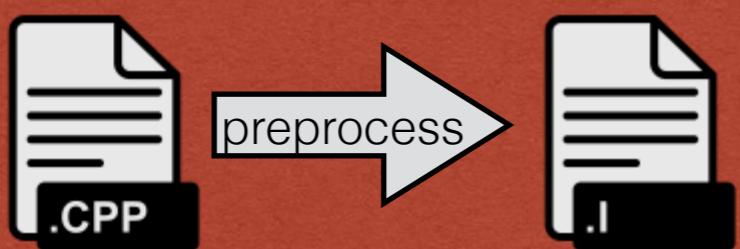
```
1 #include <vector>
2 #include "CloudBasedSuperAI.h"
```



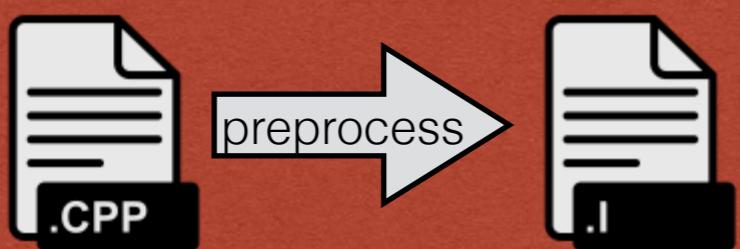




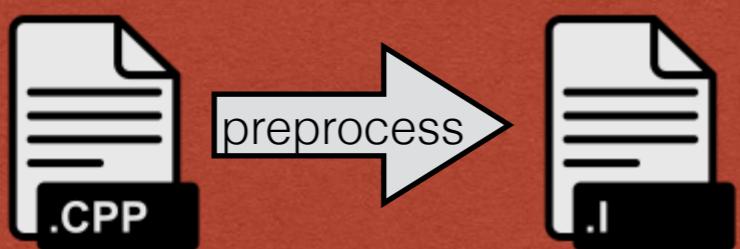
```
clang++ -E -v main.cpp -o main.i
```



```
clang++ -E -v main.cpp -o main.i
```

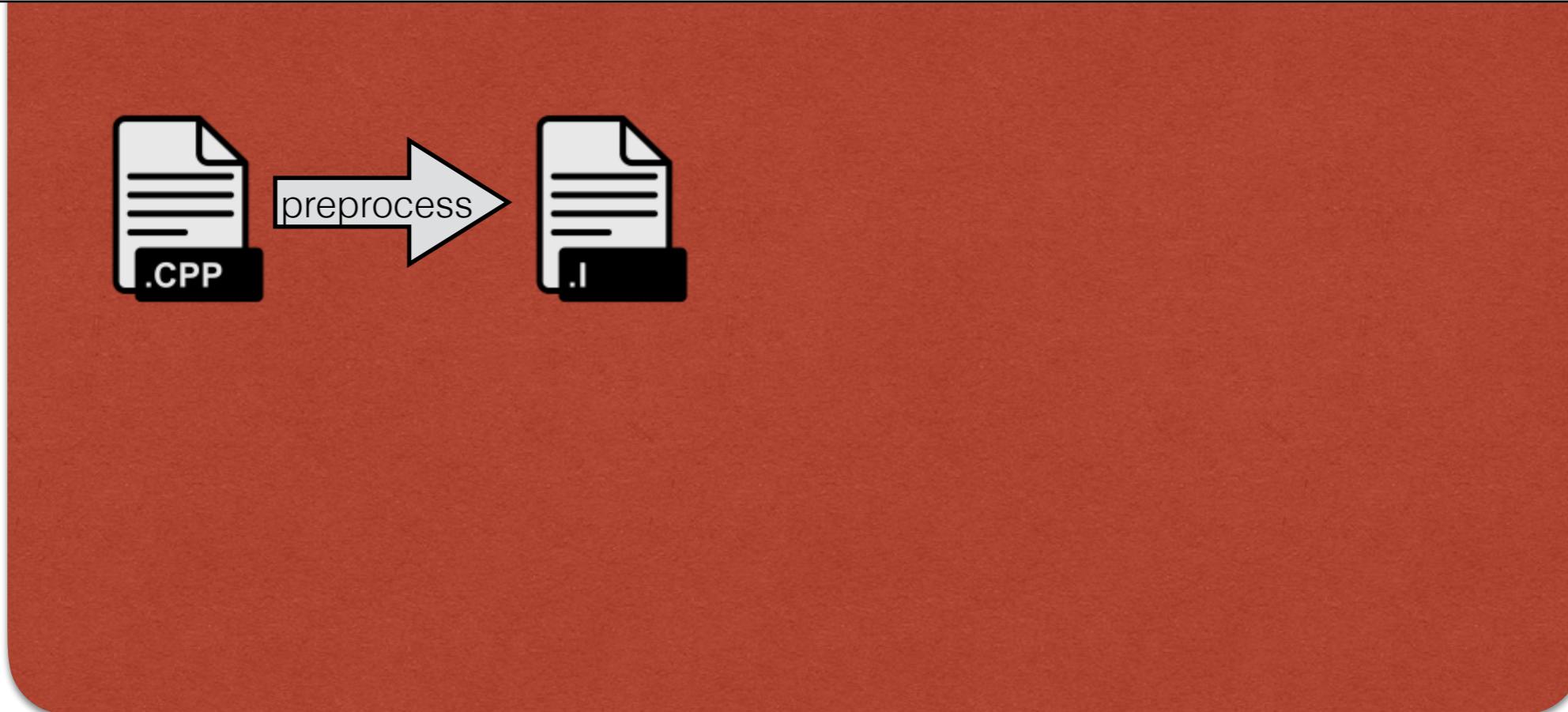


```
clang++ -E -v main.cpp -o main.i
```

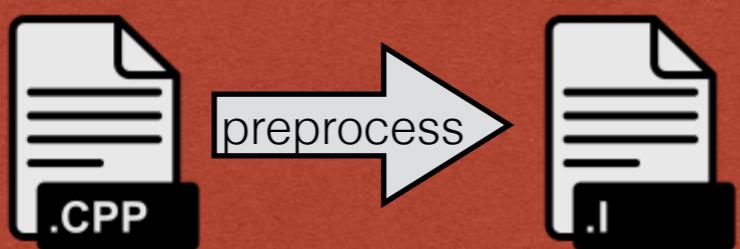


```
clang++ -E -v main.cpp -o main.i
```

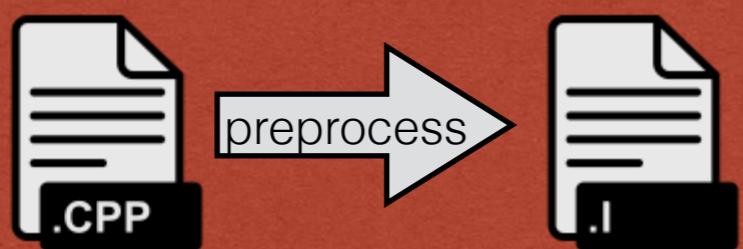
```
#include "..." search starts here:  
#include <...> search starts here:  
${ToolchainPath}/usr/bin/./include/c++/v1  
${ToolchainPath}/usr/bin/./lib/clang/8.1.0/include  
${ToolchainPath}/usr/include  
${SDKPath}/usr/include  
End of search list.
```



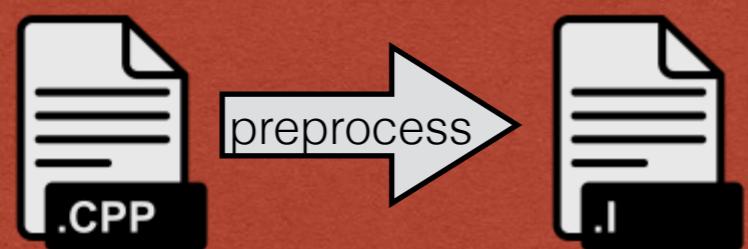
```
clang++ -E -v main.cpp -o main.i
```



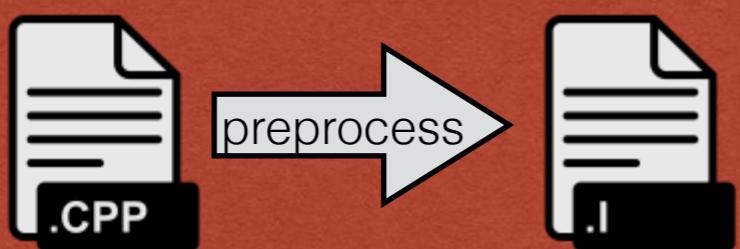
```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```



```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```

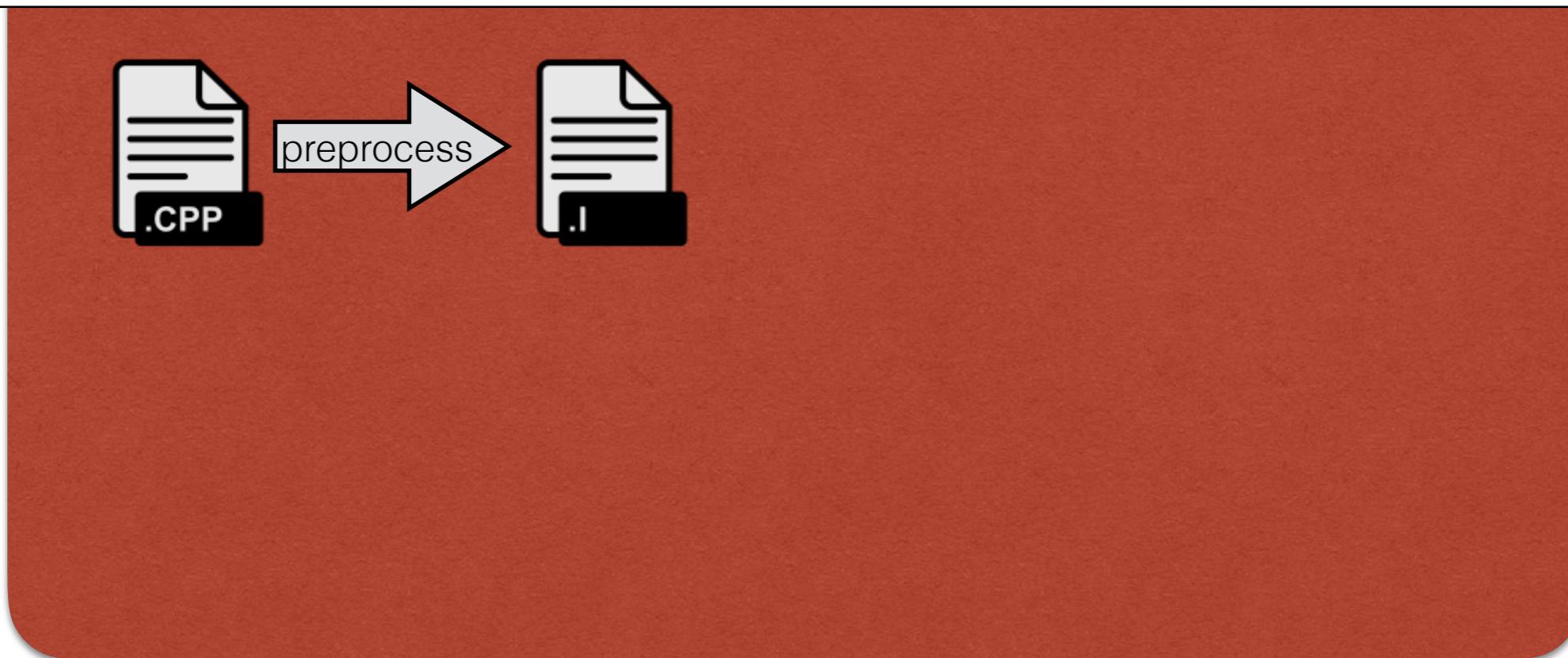


```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```



```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```

```
#include "..." search starts here:  
 ${GSL}/include  
#include <...> search starts here:  
 ${ToolchainPath}/usr/bin/./include/c++/v1  
 ${ToolchainPath}/usr/bin/./lib/clang/8.1.0/include  
 ${ToolchainPath}/usr/include  
 ${SDKPath}/usr/include  
End of search list.
```



```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```

```
#include "..." search starts here:  
 ${GSL}/include  
#include <...> search starts here:  
 ${ToolchainPath}/usr/bin/.../include/c++/v1  
 ${ToolchainPath}/usr/bin/.../lib/clang/8.1.0/include  
 ${ToolchainPath}/usr/include  
 ${SDKPath}/usr/include
```

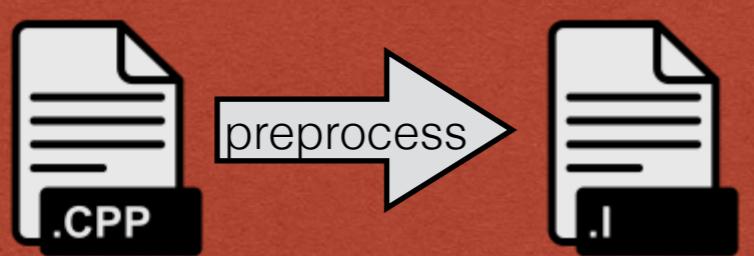
End of search list.

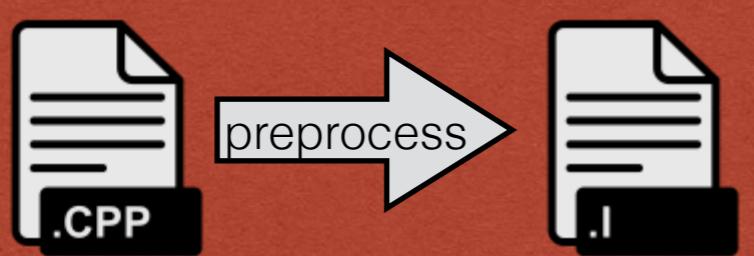


```
clang++ -E -v -iquote ${GSL}/include main.cpp -o main.i
```

```
#include "..." search starts here:  
 ${GSL}/include  
#include <...> search starts here:  
 ${ToolchainPath}/usr/bin/./include/c++/v1  
 ${ToolchainPath}/usr/bin/./lib/clang/8.1.0/include  
 ${ToolchainPath}/usr/include  
 ${SDKPath}/usr/include  
End of search list.
```

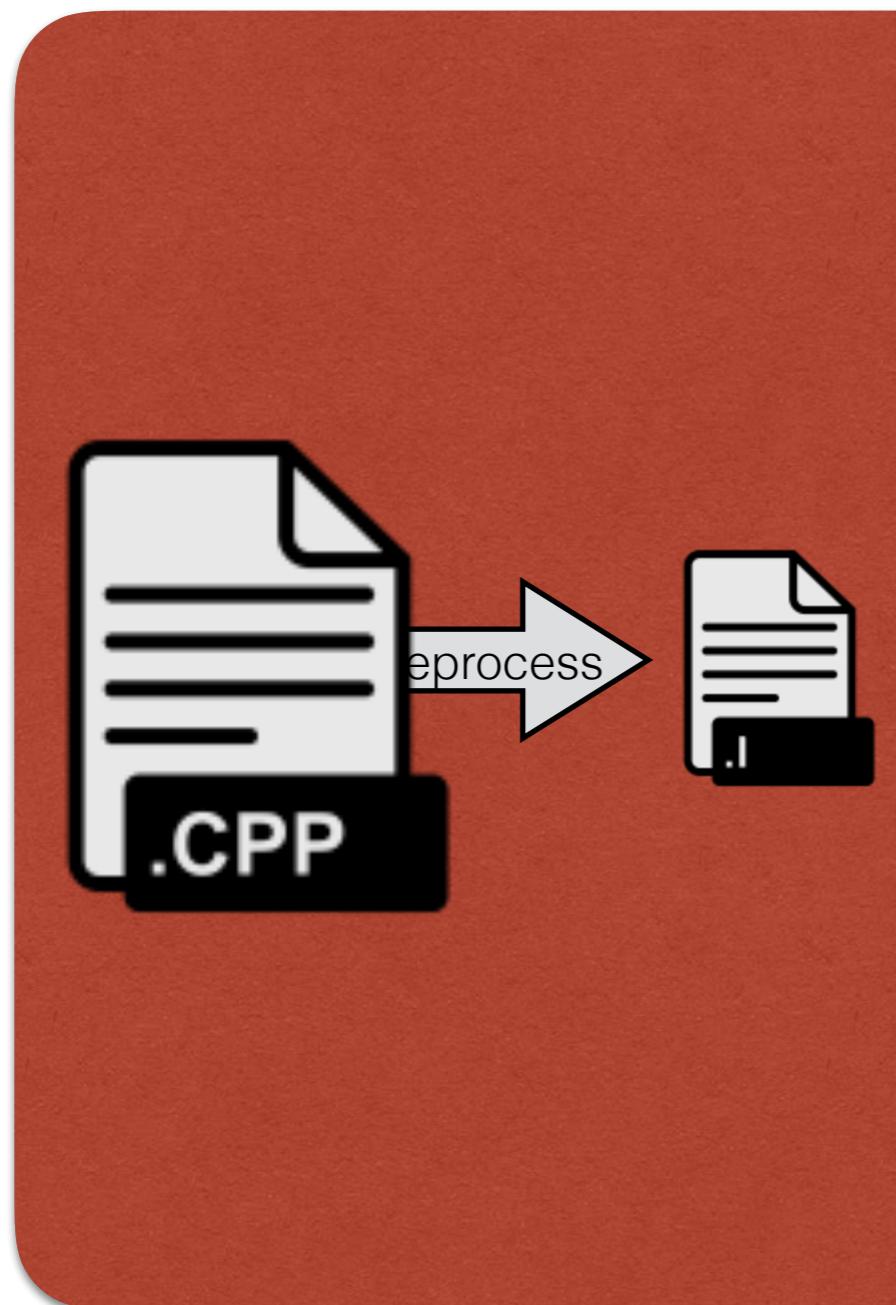


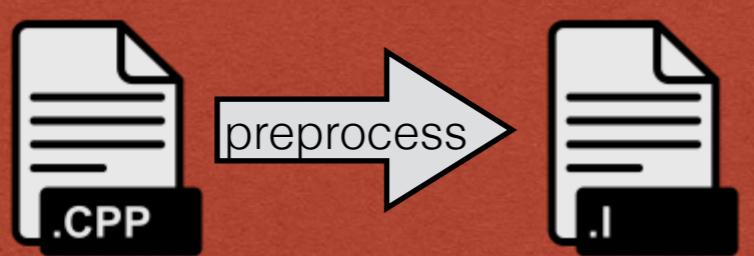


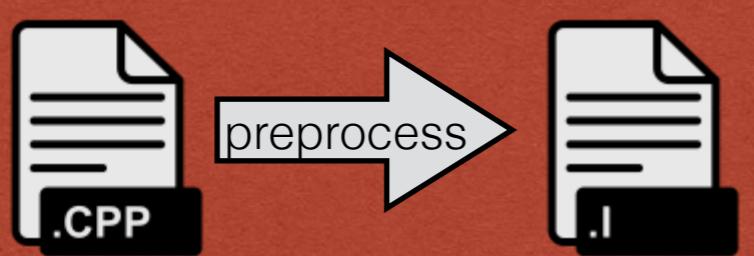


main.cpp

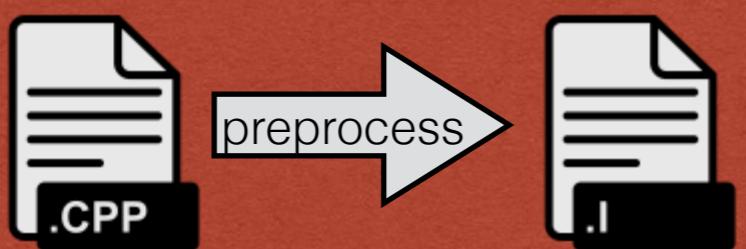
```
1 #include "CloudBasedSuperAI.h"
2
3 int main()
4 {
5 #if DEBUG
6     log("Debug configuration");
7#endif
8     runCloudBasedSuperAI();
9     return 0;
10}
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```



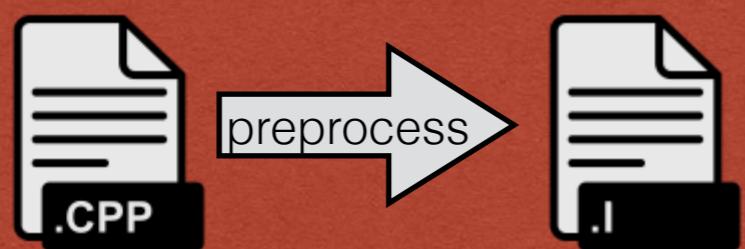




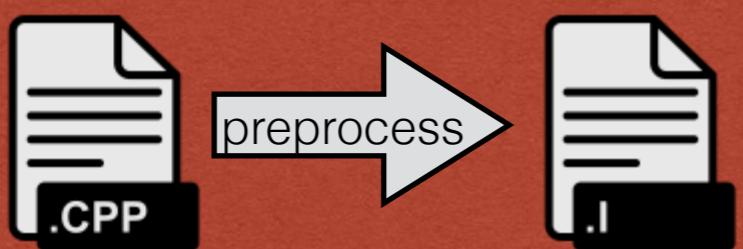
```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```



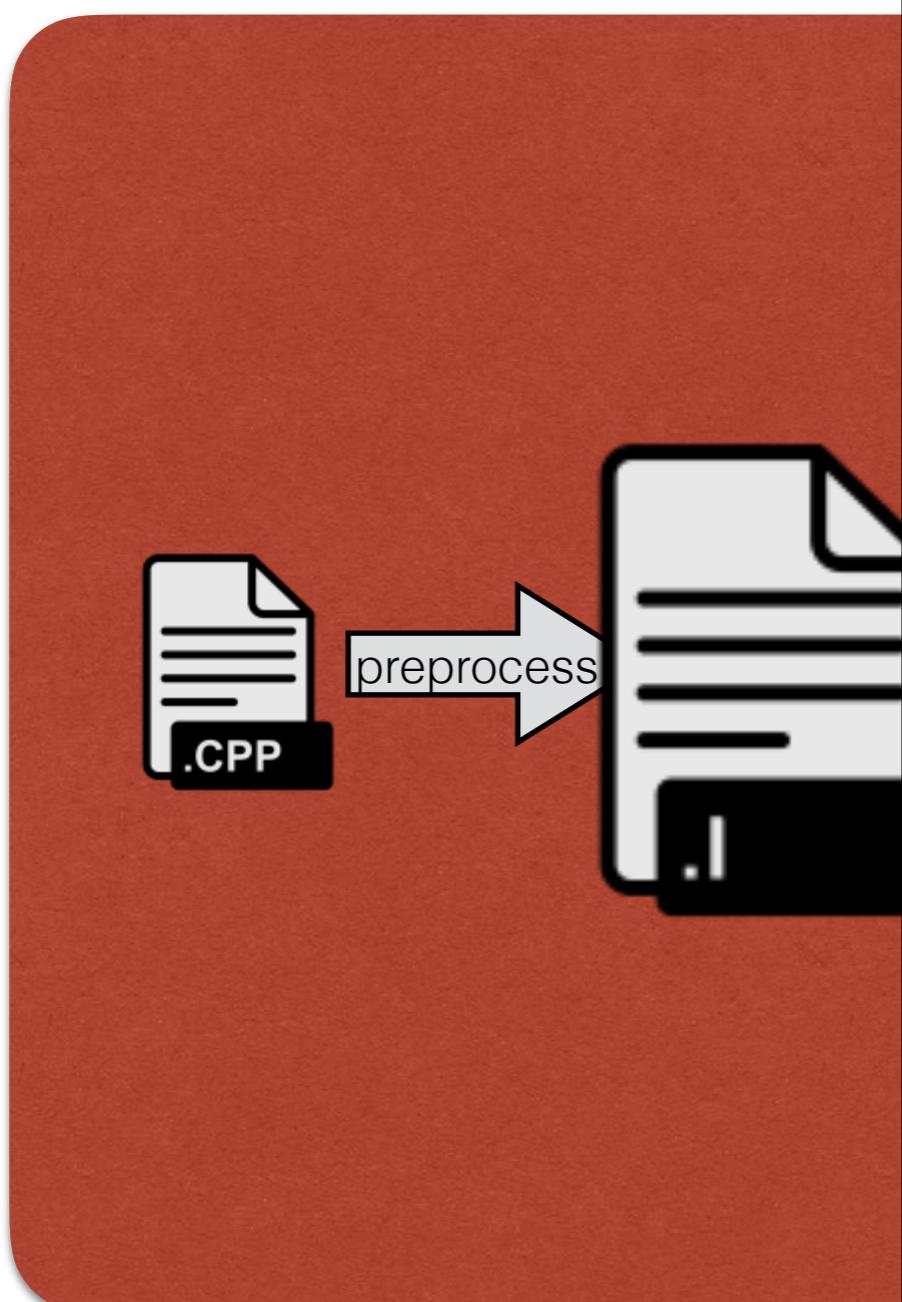
```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```



```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```



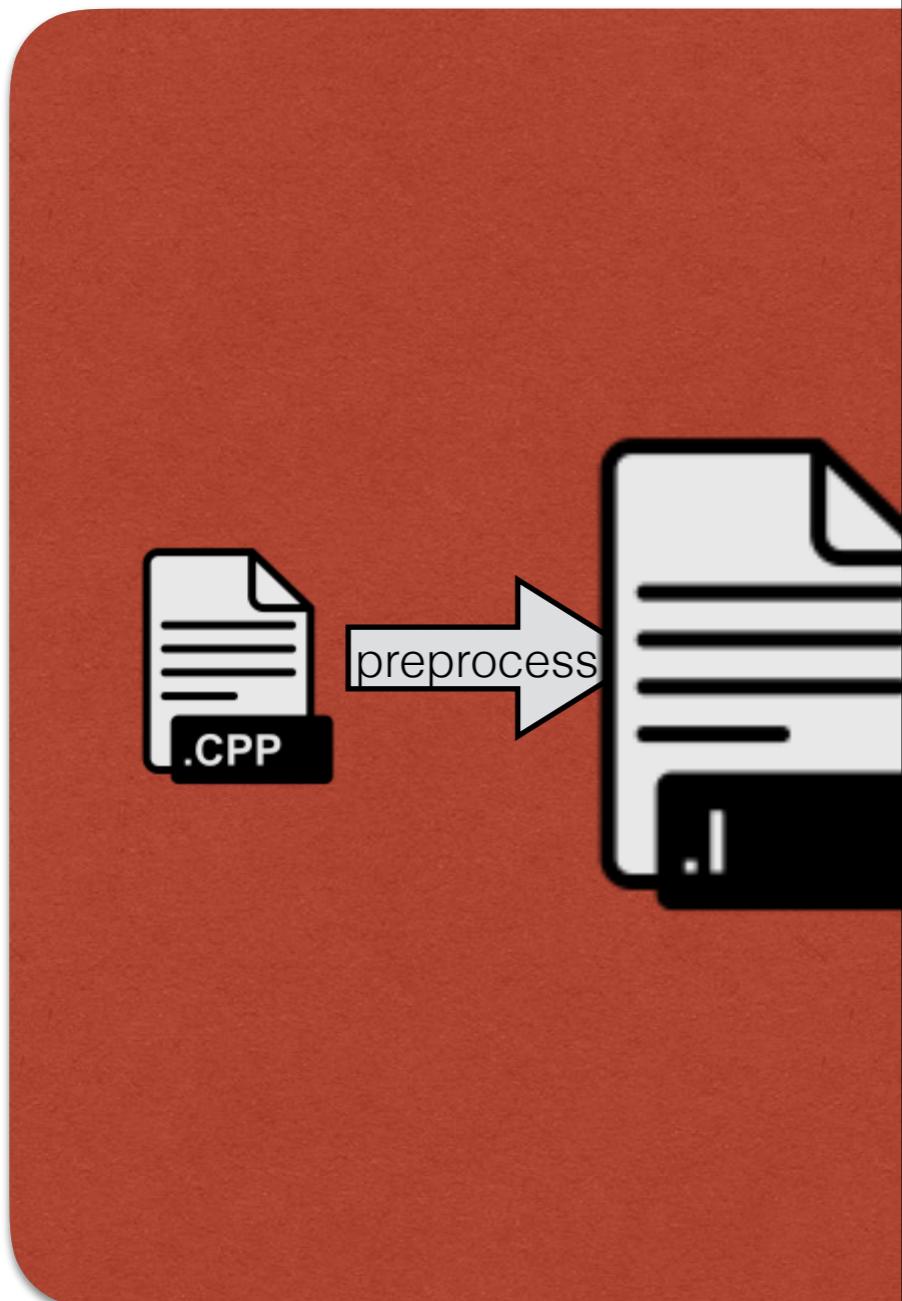
```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```



main.i

```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17
18     log("Debug configuration");
19
20     runCloudBasedSuperAI();
21     return 0;
22 }
```

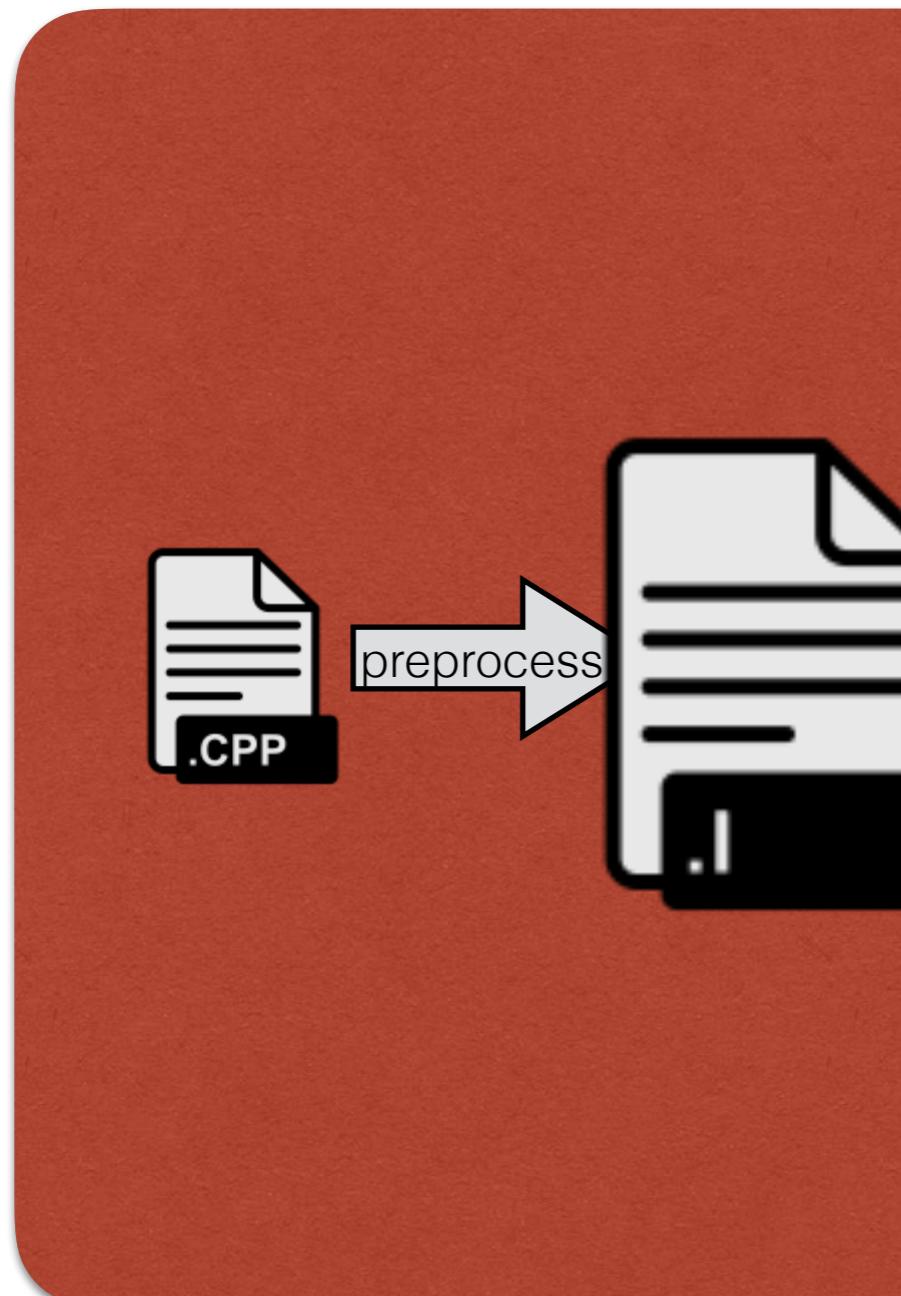
```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```



main.i

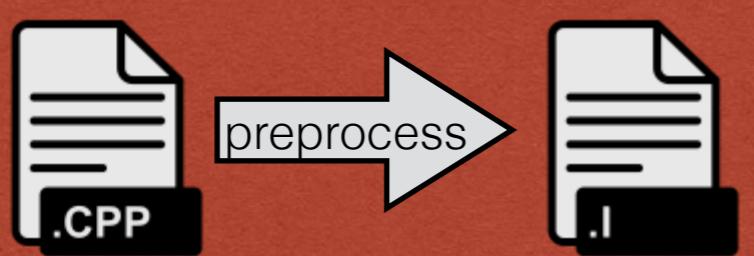
```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17
18     log("Debug configuration");
19
20     runCloudBasedSuperAI();
21     return 0;
22 }
```

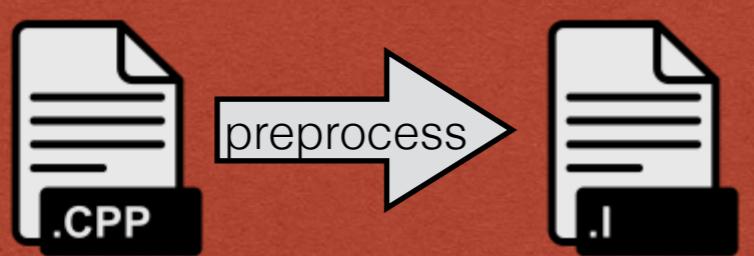
```
clang++ -E -D DEBUG=1 main.cpp -o main.i
```

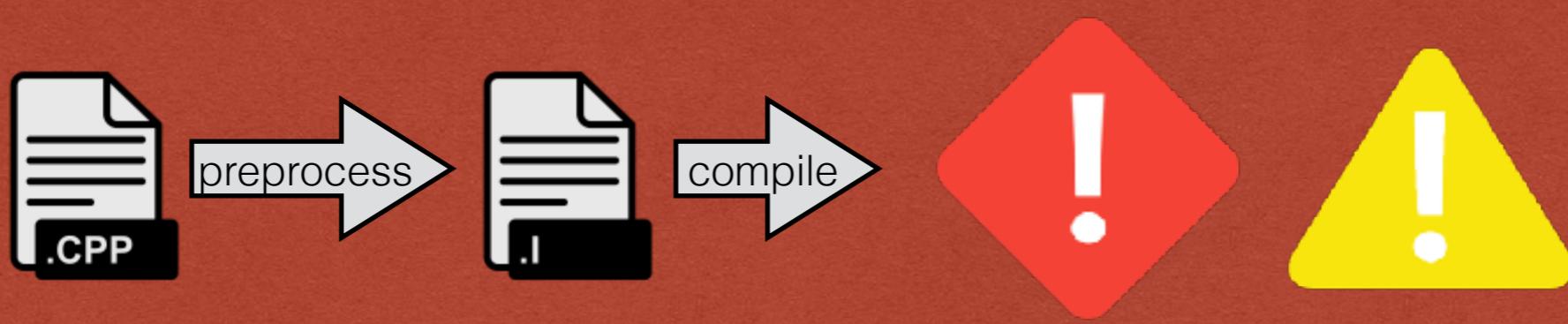


main.i

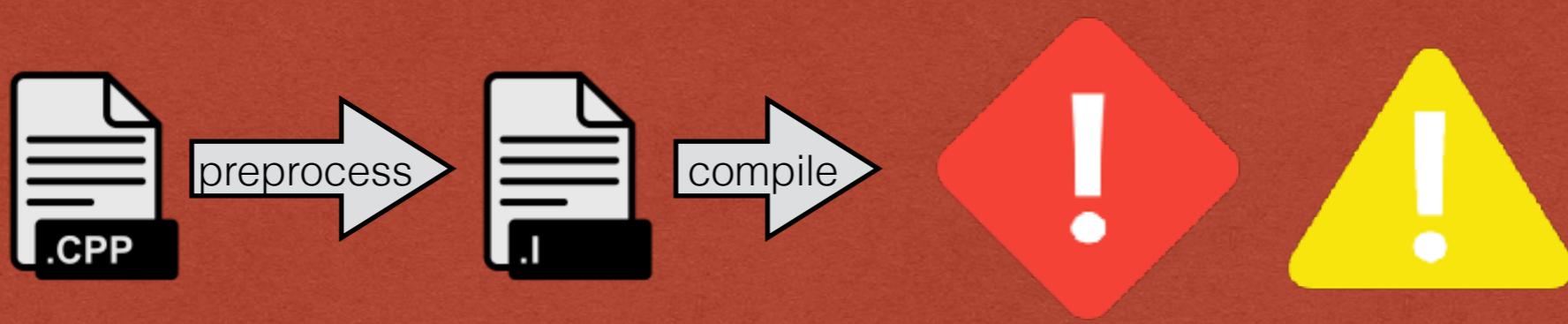
```
1 # 1 "main.cpp"
2 # 1 "<built-in>" 1
3 # 1 "<built-in>" 3
4 # 340 "<built-in>" 3
5 # 1 "<command line>" 1
6 # 1 "<built-in>" 2
7 # 1 "main.cpp" 2
8 # 1 "./CloudBasedSuperAI.h" 1
9
10
11
12 void runCloudBasedSuperAI();
13 # 2 "main.cpp" 2
14
15 int main()
16 {
17
18     log("Debug configuration");
19
20     runCloudBasedSuperAI();
21     return 0;
22 }
```

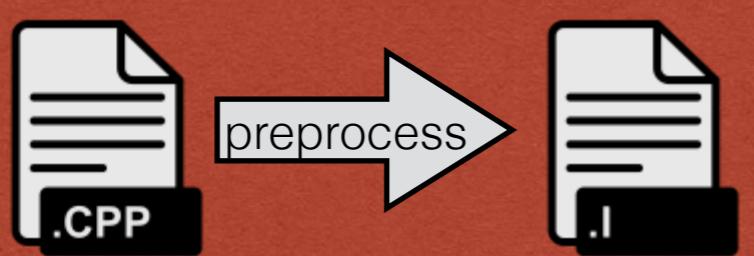
















```
clang++ -S main.cpp -o main.s
```



```
clang++ -S main.cpp -o main.s
```



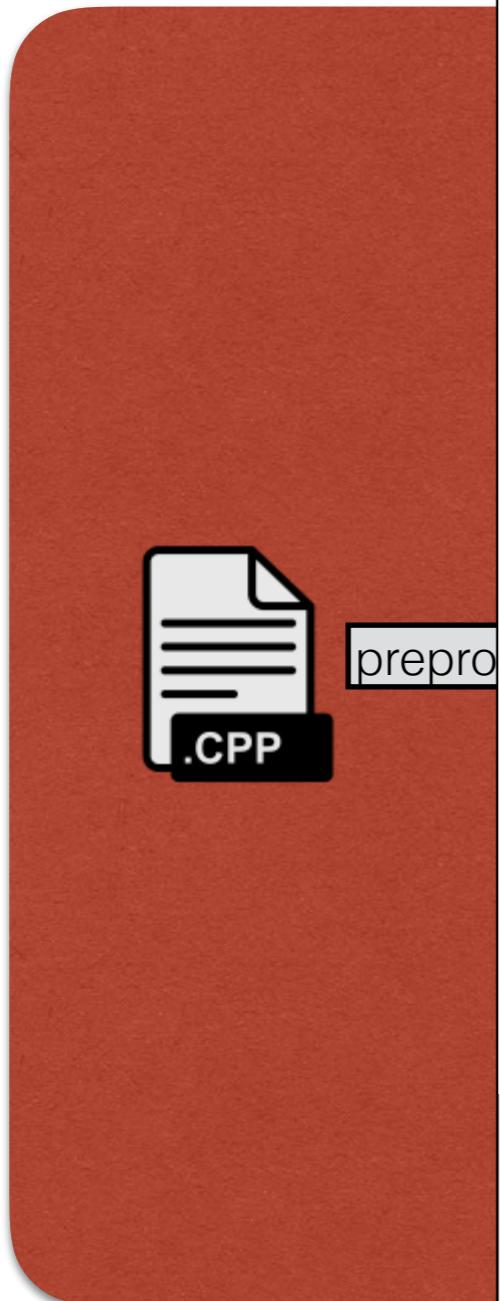
```
clang++ -S main.cpp -o main.s
```



```
clang++ -S main.cpp -o main.s
```



```
clang++ -S main.cpp -o main.s
```

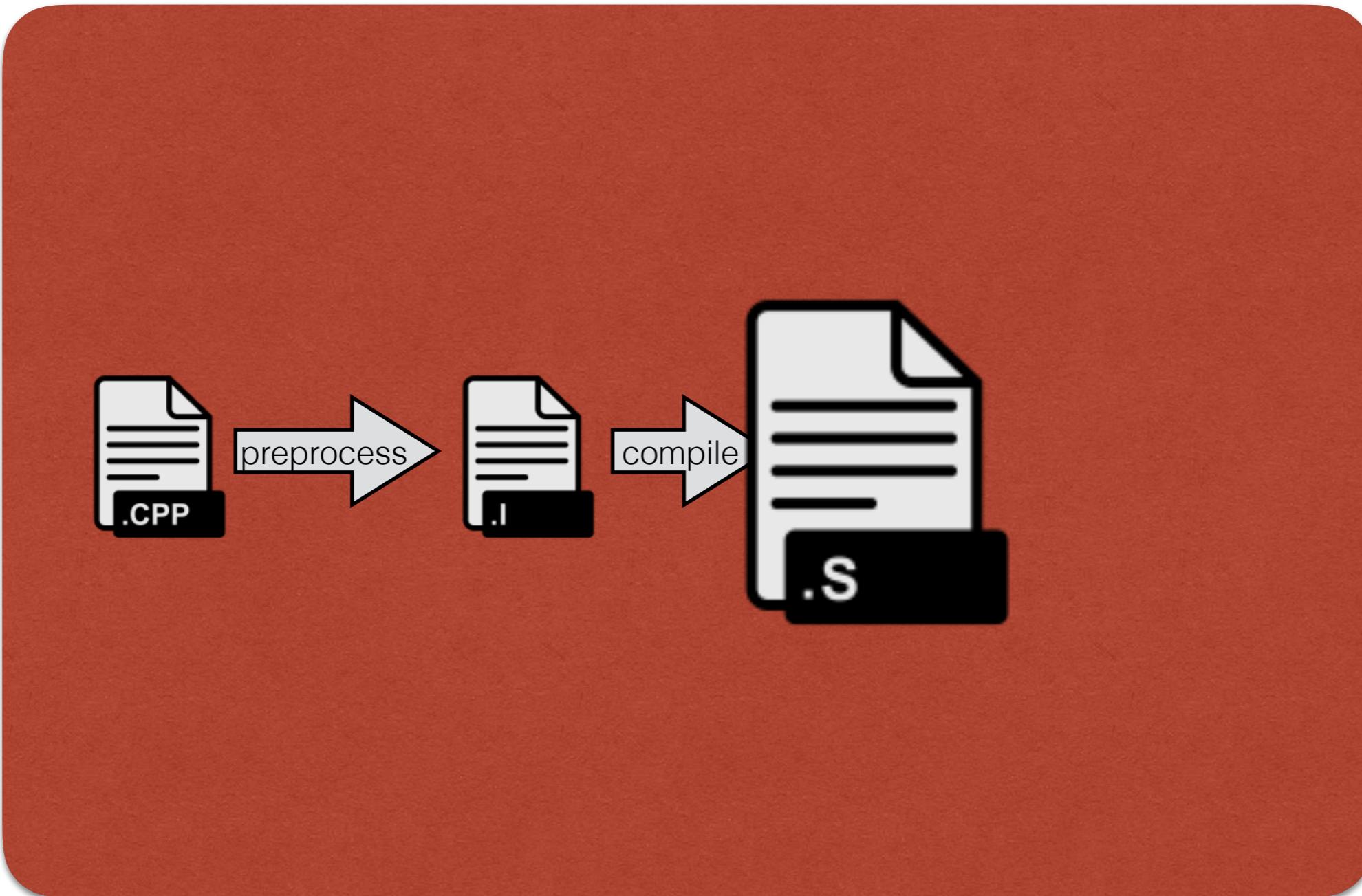


main.s

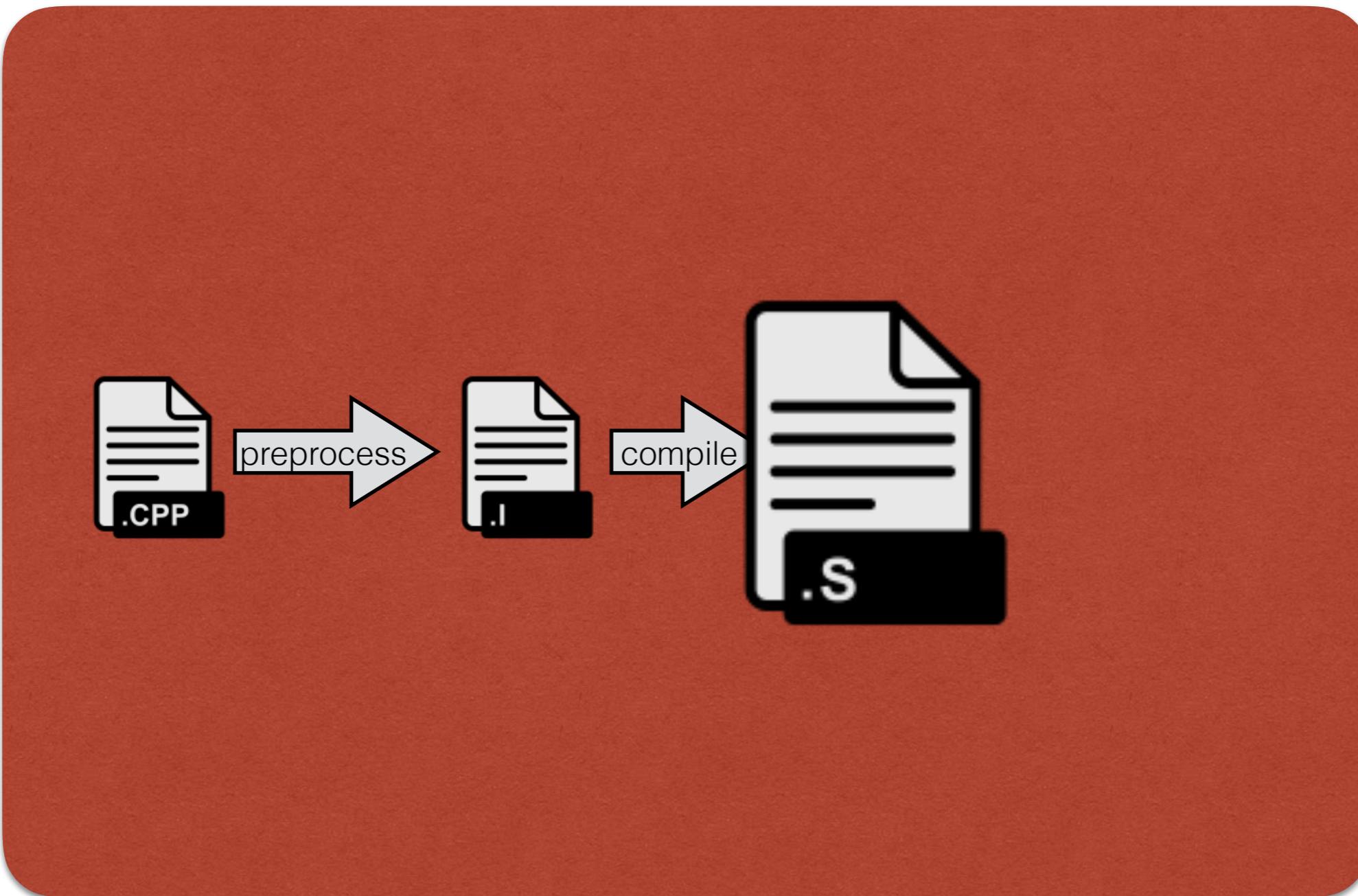
```
1 main:  
2     pushq   %rbp  
3     movq    %rsp, %rbp  
4     subq    $16, %rsp  
5     movl    $0, -4(%rbp)  
6     callq   runCloudBasedSuperAI()  
7     xorl    %eax, %eax  
8     addq    $16, %rsp  
9     popq    %rbp  
10    retq
```



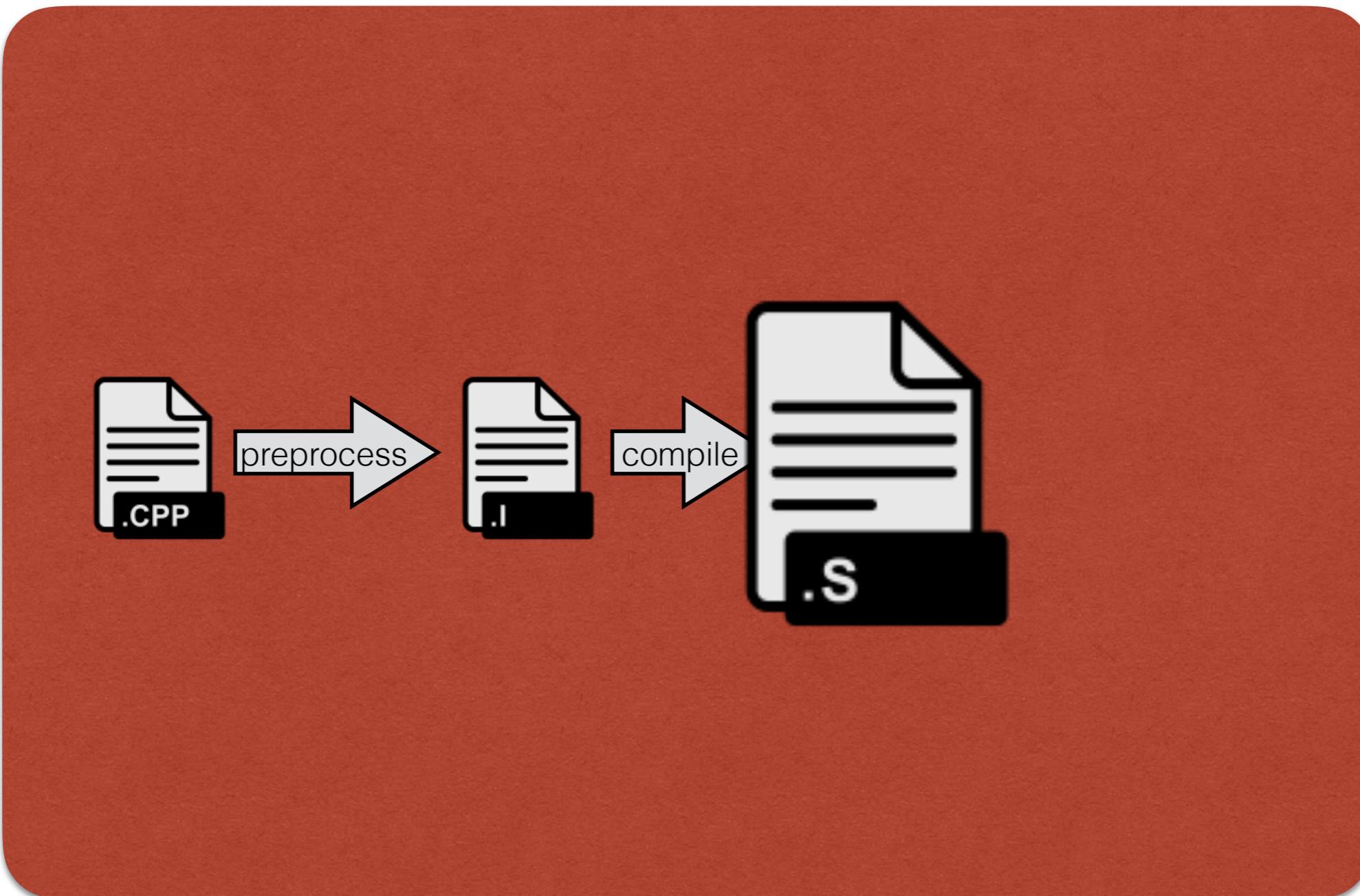
```
clang++ -S main.cpp -o main.s
```



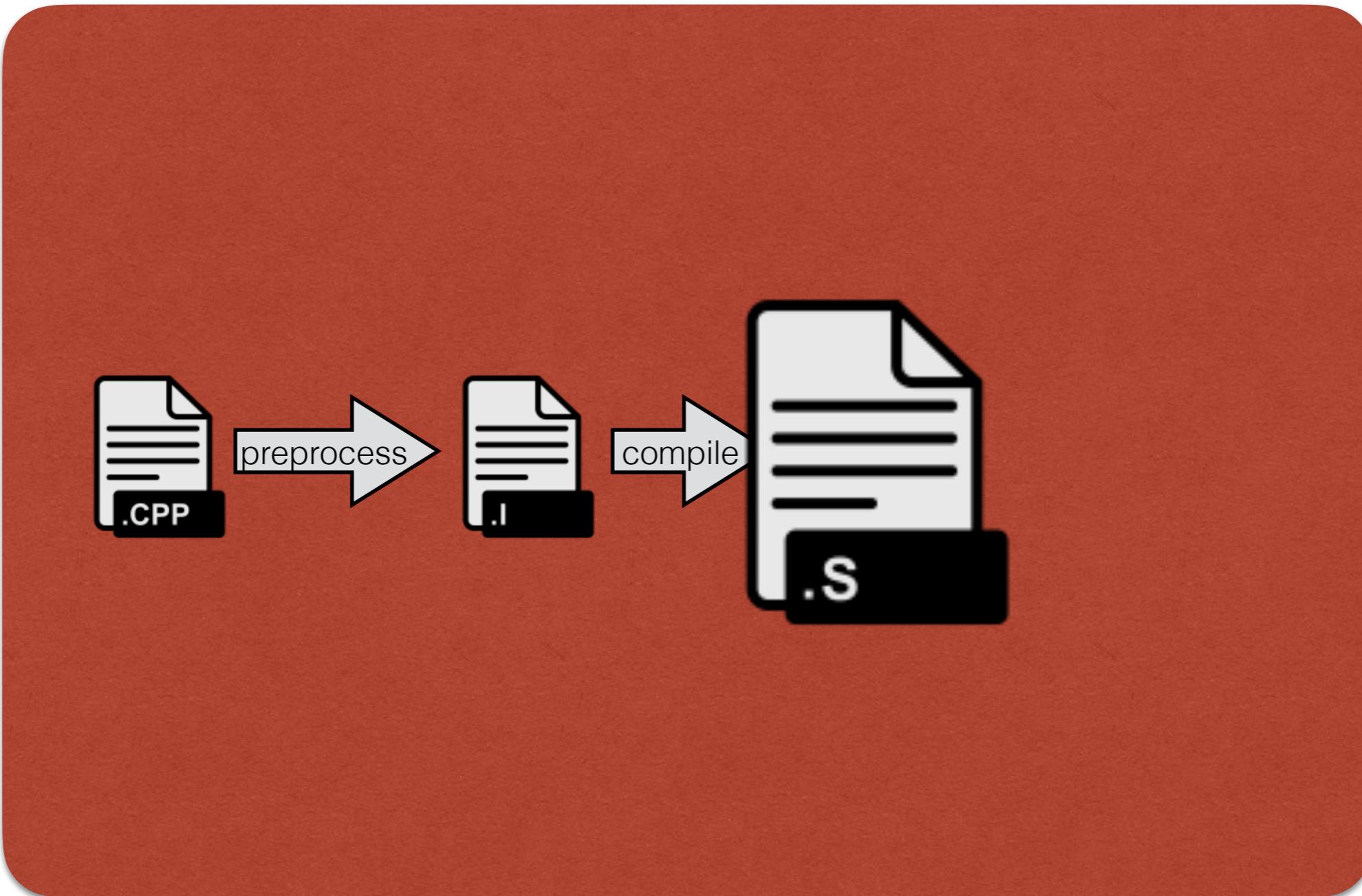
```
clang++ -S -O3 main.cpp -o main.s
```



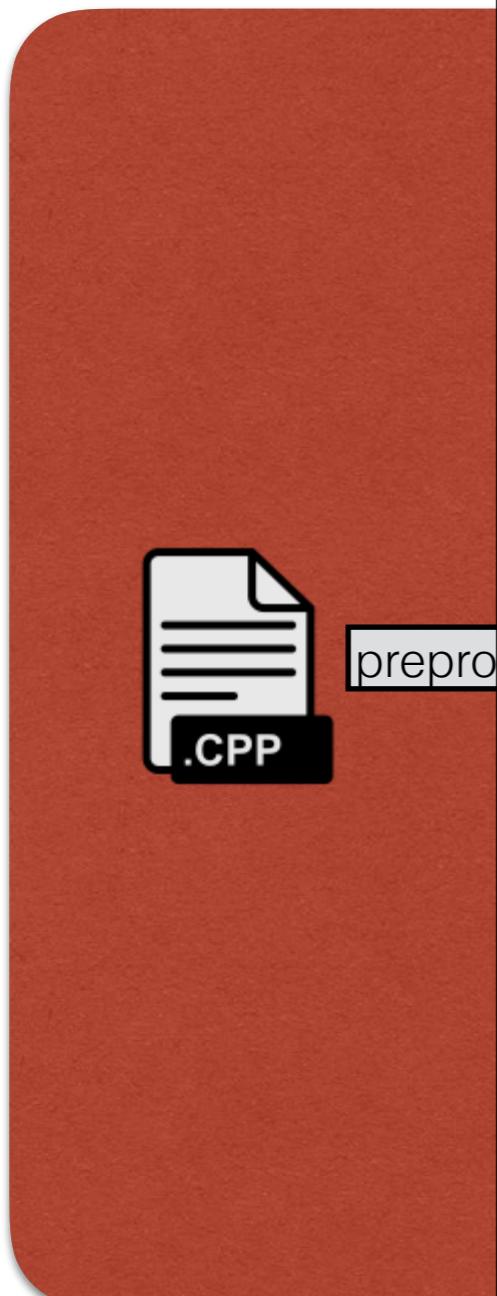
```
clang++ -S -O3 main.cpp -o main.s
```



```
clang++ -S -O3 main.cpp -o main.s
```



```
clang++ -S -O3 main.cpp -o main.s
```

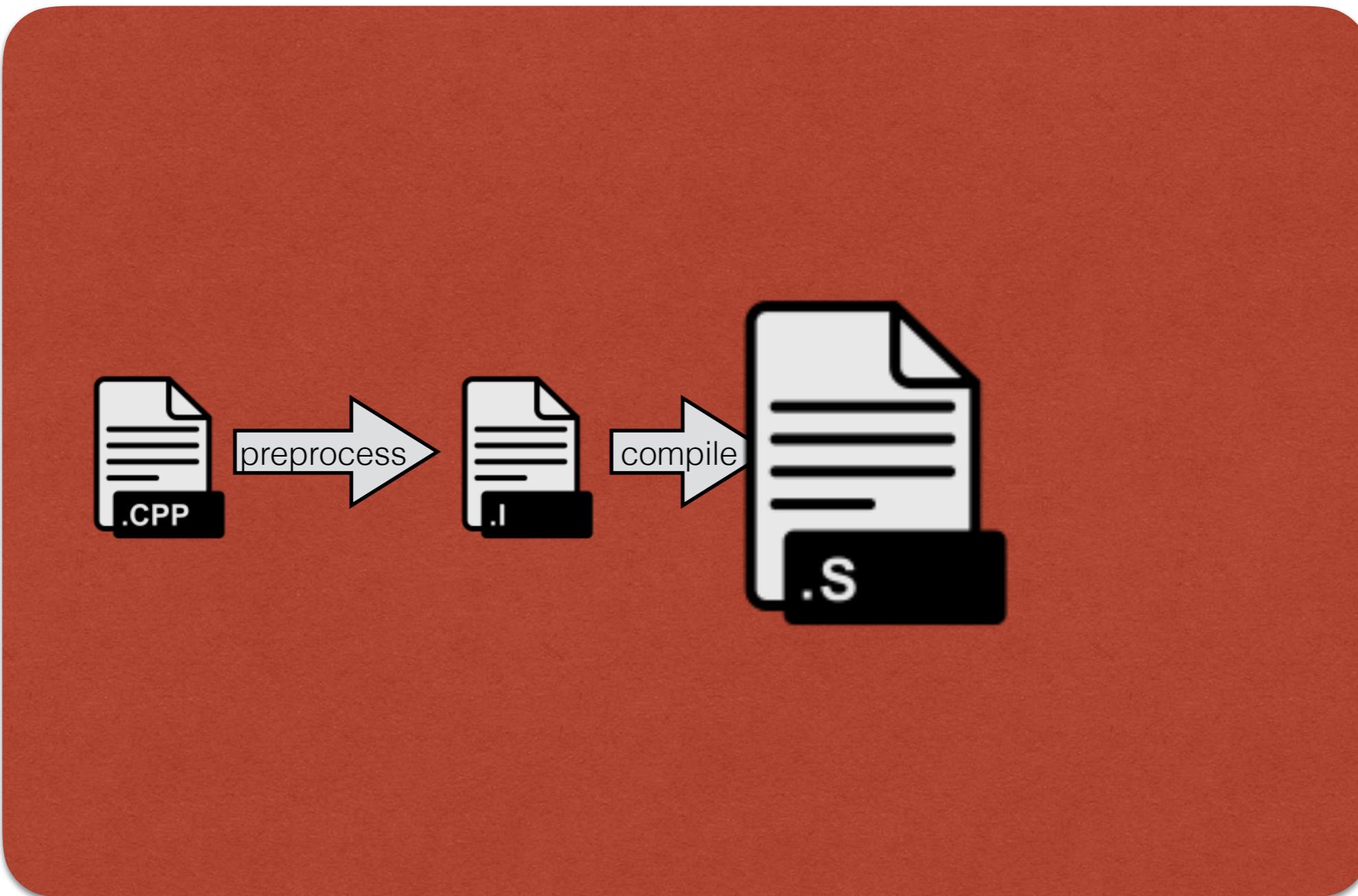


main.s

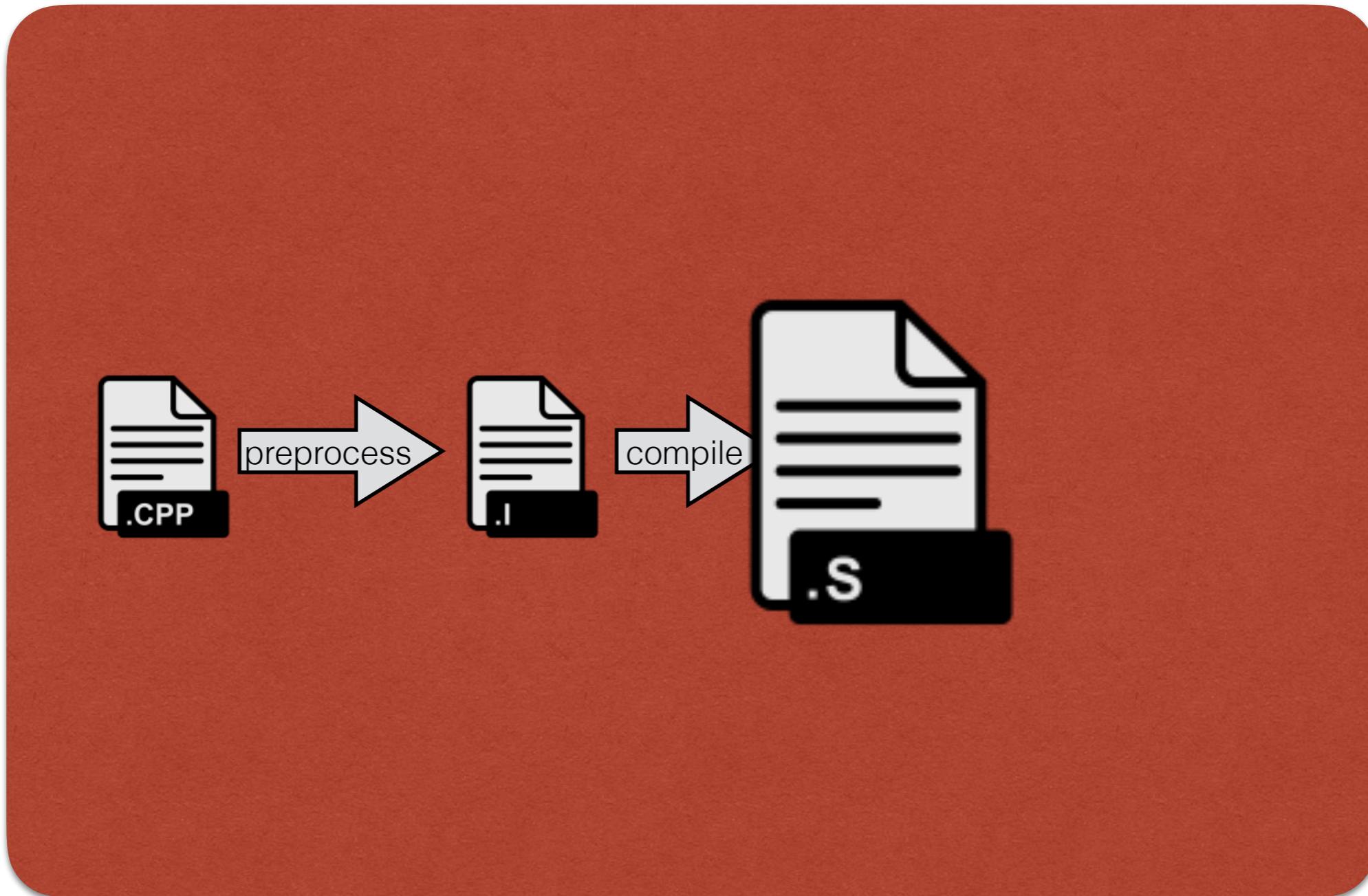
```
1 main:  
2     pushq   %rax  
3     callq   runCloudBasedSuperAI()  
4     xorl    %eax, %eax  
5     popq    %rcx  
6     retq
```



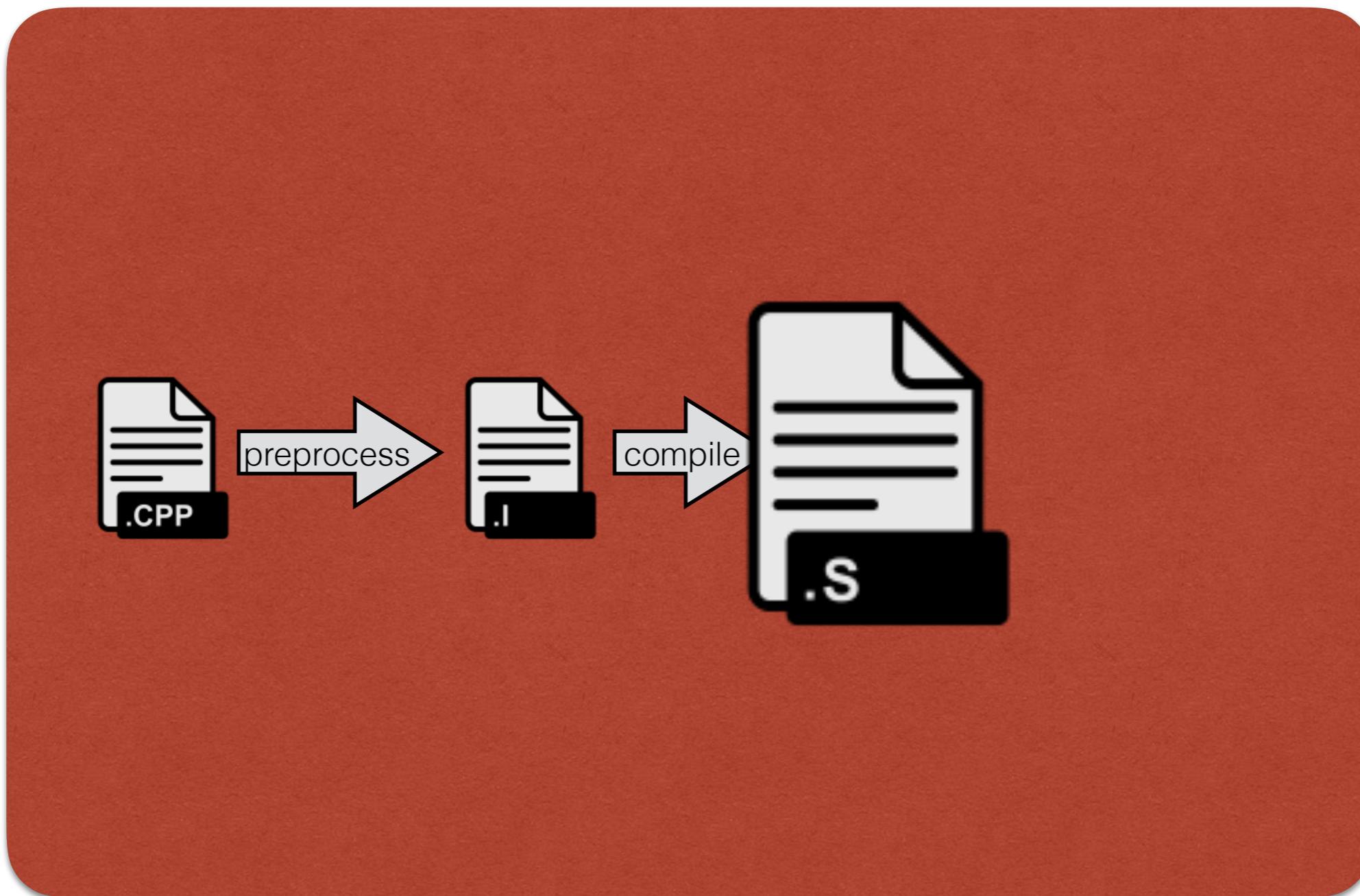
```
clang++ -S -O3 main.cpp -o main.s
```



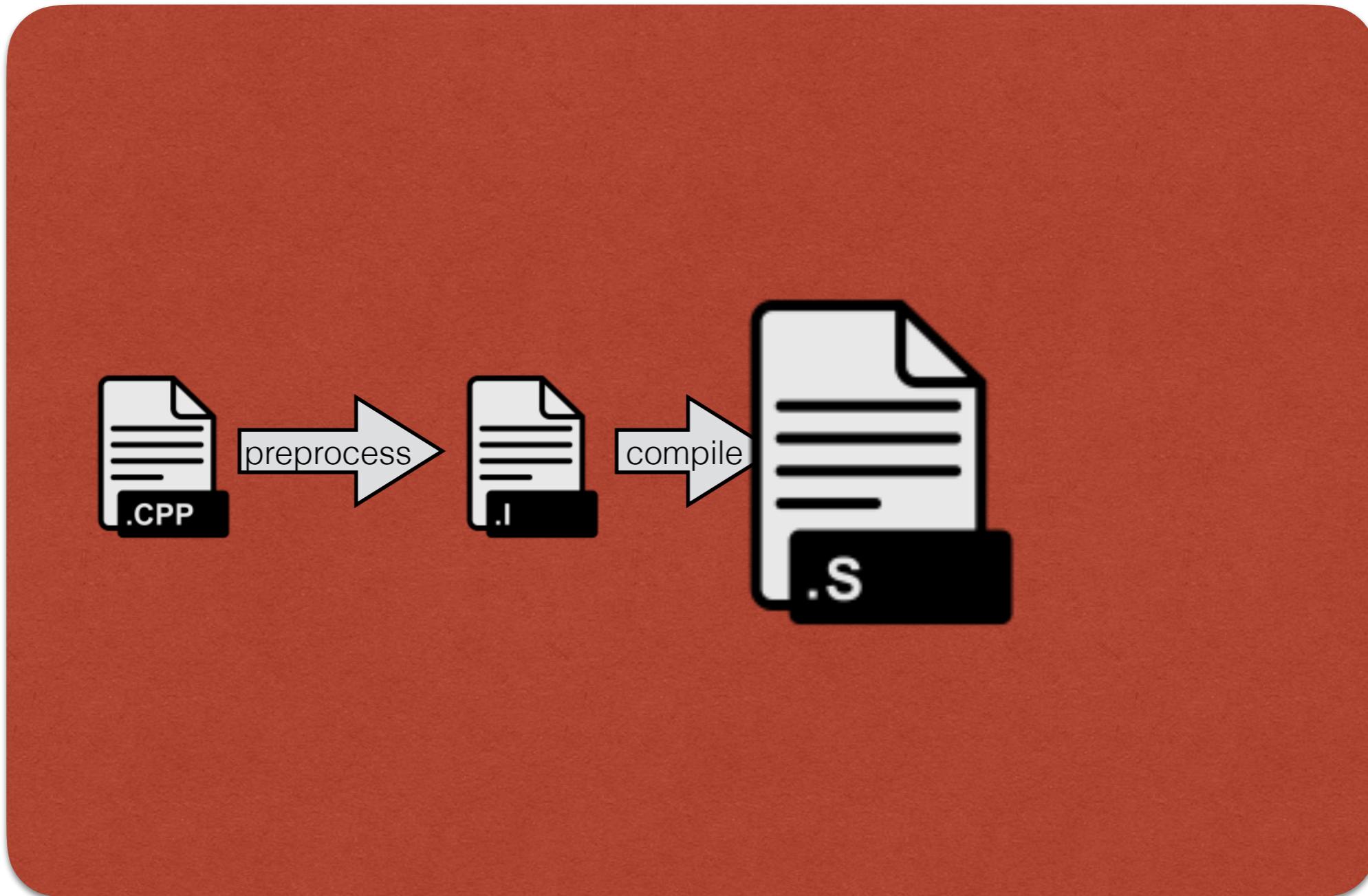
```
clang++ -S -O3 -fsanitize=address main.cpp -o main.s
```



```
clang++ -S -O3 -fsanitize=address main.cpp -o main.s
```



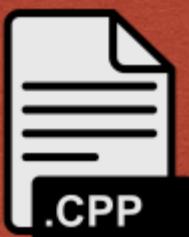
```
clang++ -S -O3 -fsanitize=address main.cpp -o main.s
```



```
clang++ -S -O3 -fsanitize=address main.cpp -o main.s
```

main.s

```
1 main:                                     # @main
2     pushq   %rax
3     callq   runCloudBasedSuperAI()
4     xorl   %eax, %eax
5     popq   %rcx
6     retq
7
8 asan.module_ctor:                         # @asan.module_ctor
9     pushq   %rax
10    callq   __asan_init
11    popq   %rax
12    jmp    __asan_version_mismatch_check_v8 # TAILCALL
```

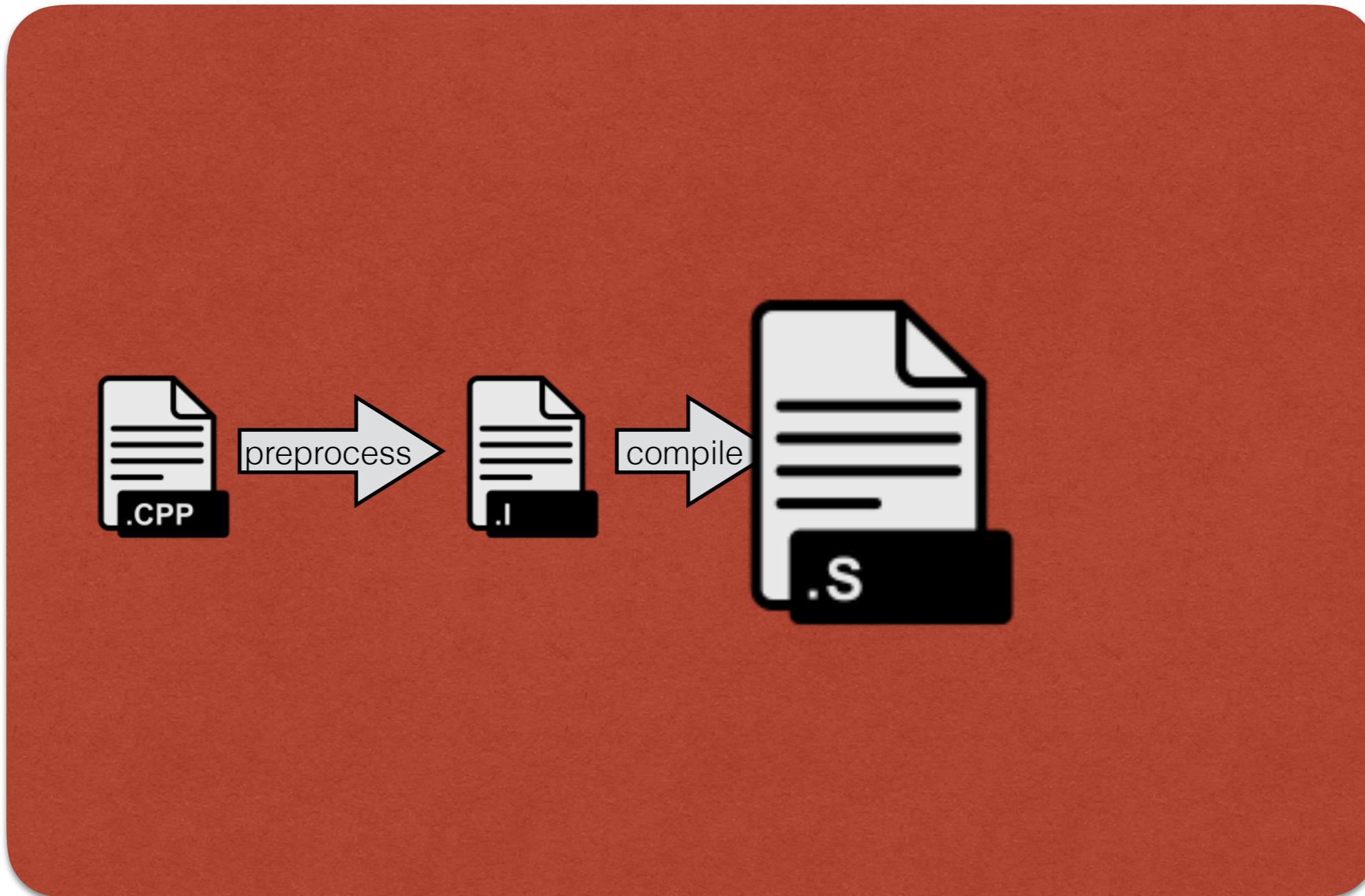


prepro

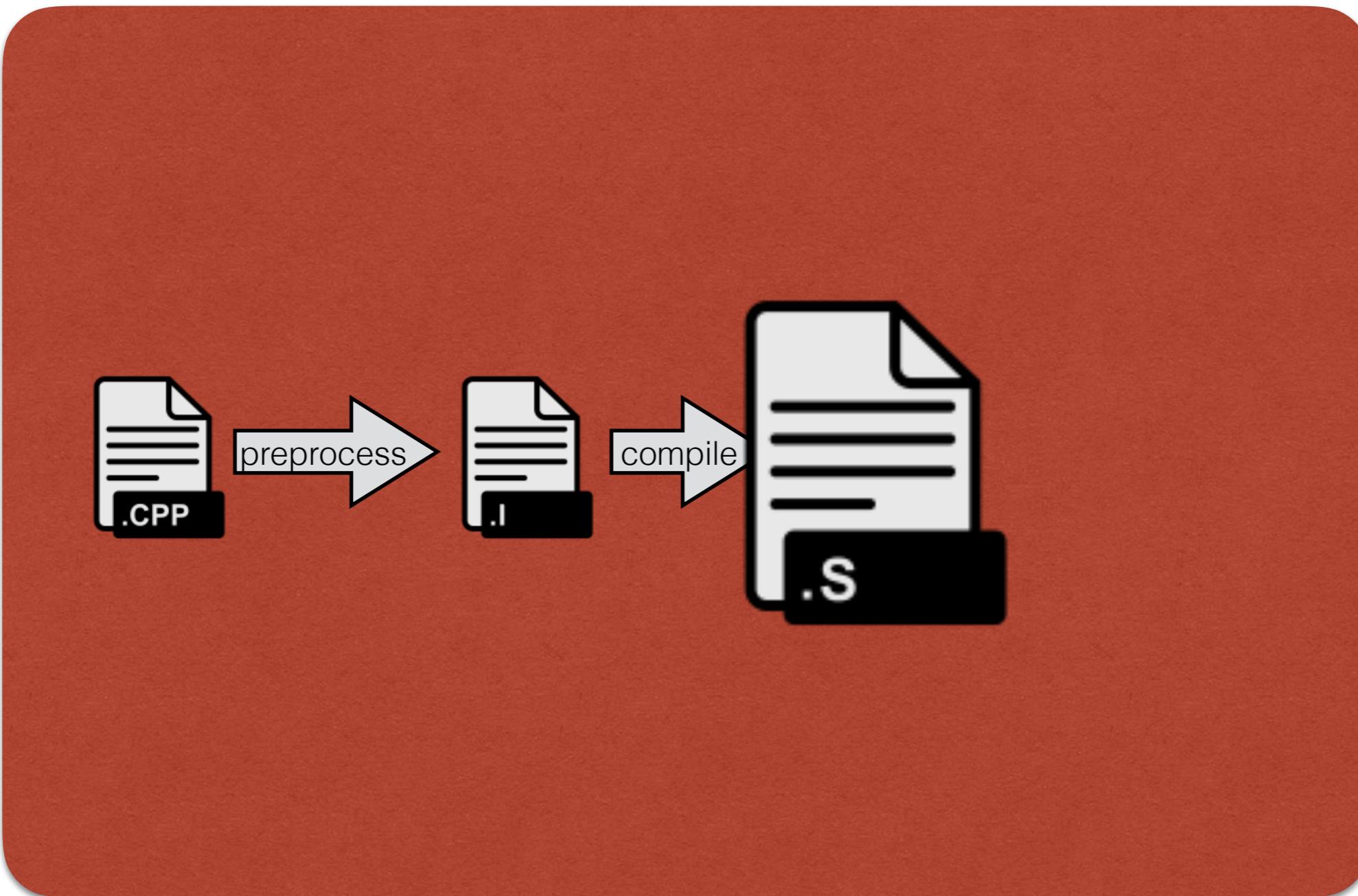


<https://gcc.godbolt.org/>

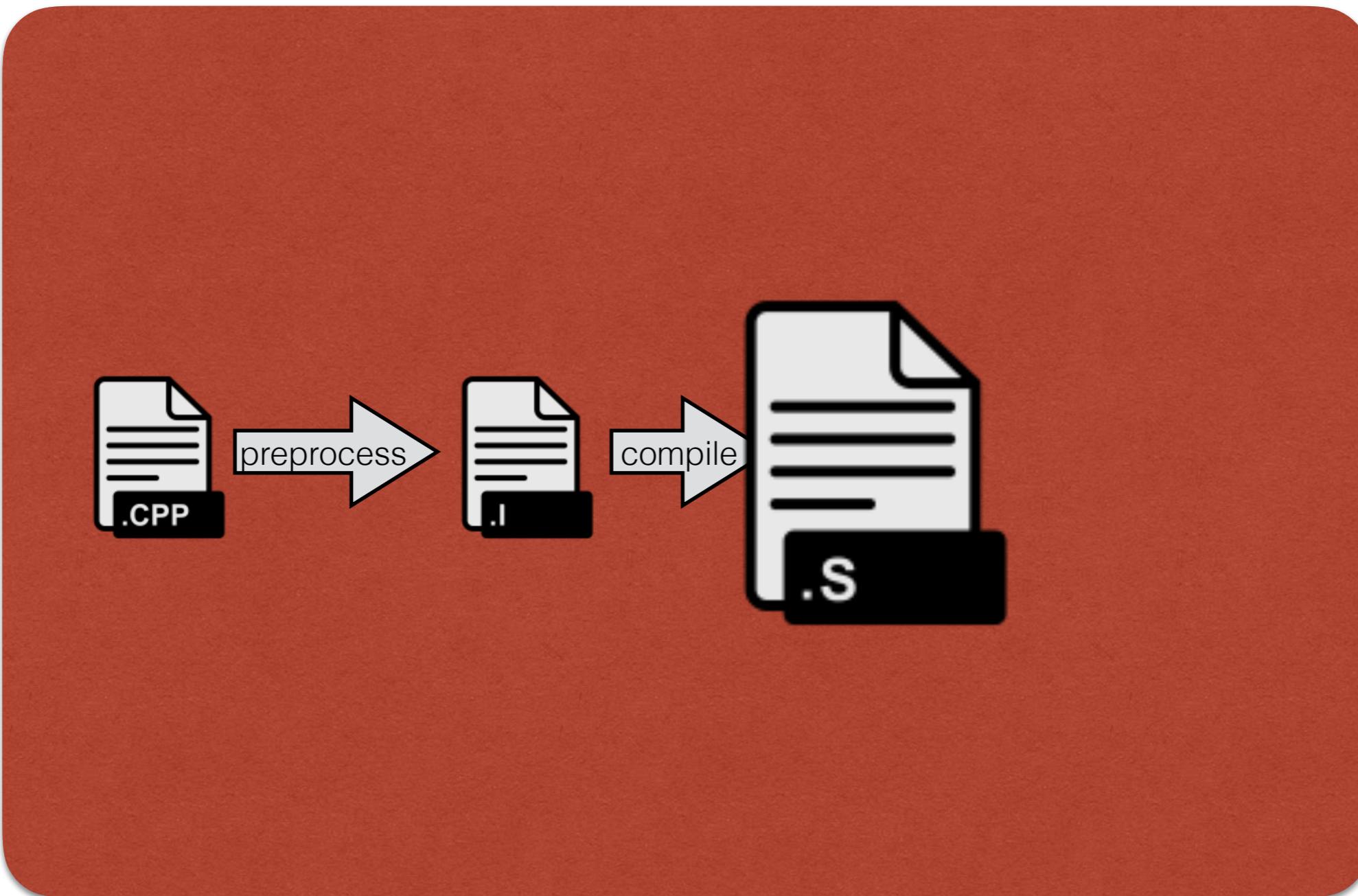
```
clang++ -S -O3 -fsanitize=address main.cpp -o main.s
```



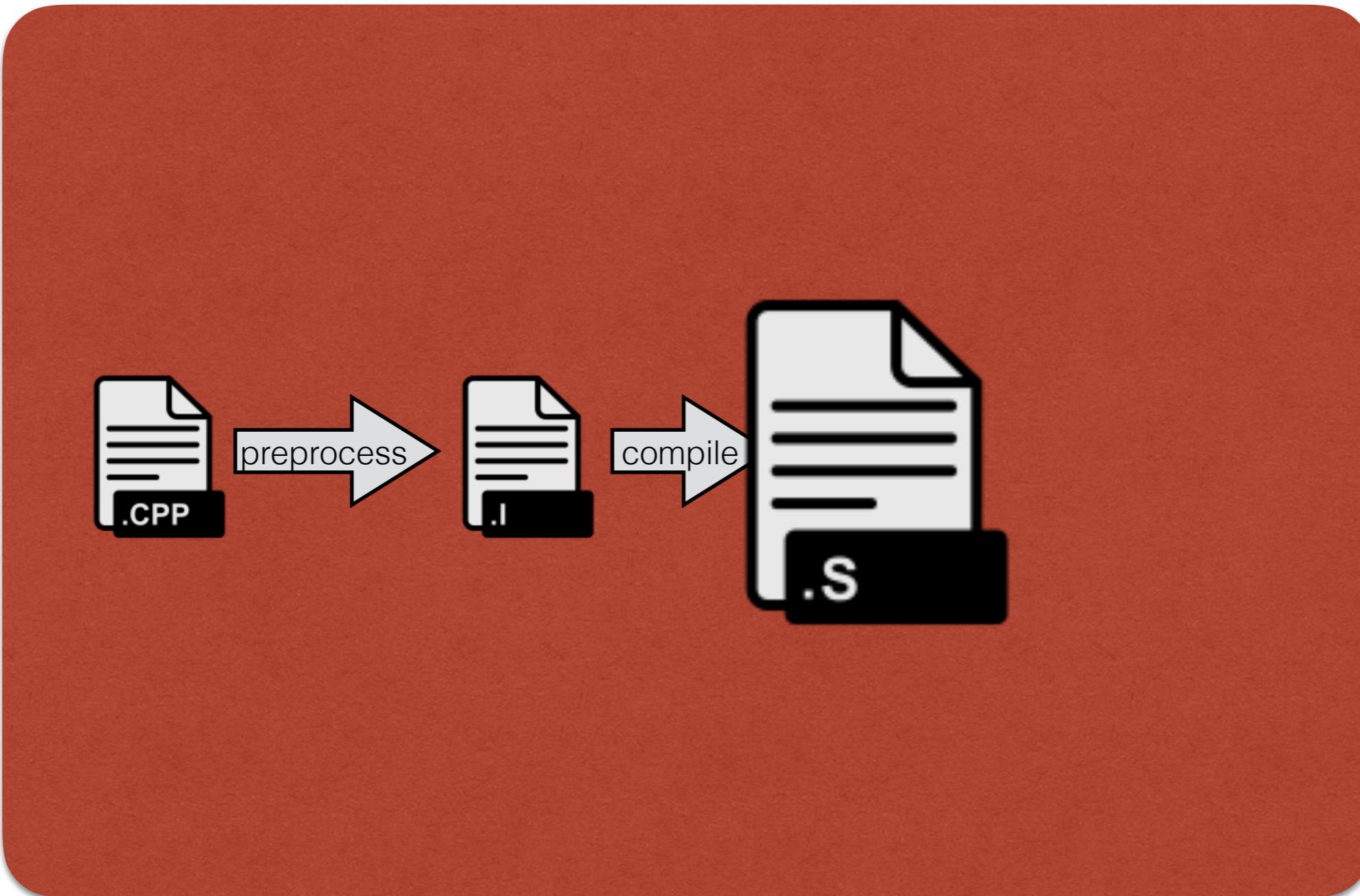
```
clang++ -S -g main.cpp -o main.s
```



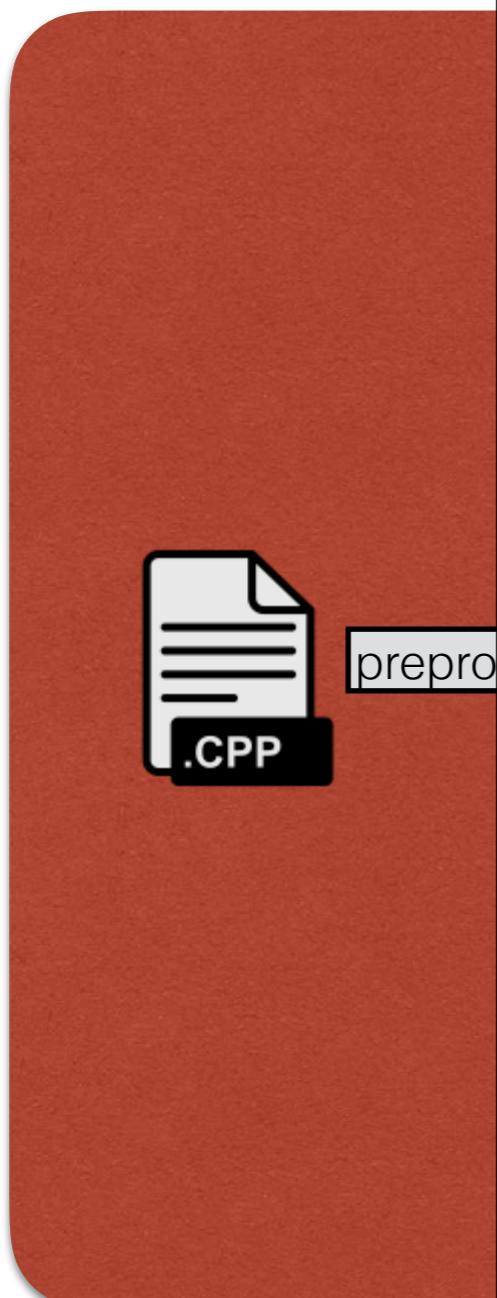
```
clang++ -S -g main.cpp -o main.s
```



```
clang++ -S -g main.cpp -o main.s
```



```
clang++ -S -g main.cpp -o main.s
```



main.s

```
1      .text
2      .file  "/tmp/compiler-explorer-compiler117427-5
3      .globl main
4      .p2align    4, 0x90
5      .type   main,@function
6 main:                                # @main
7 .Lfunc_begin0:
8     .file  1 "/tmp/compiler-explorer-compiler117427-5
9     .loc   1 4 0                      # /tmp/compiler-
10    .cfi_startproc
11    .loc   1 5 5 prologue_end        # /tmp/compiler-
12    pushq  %rax
13    .cfi_def_cfa_offset 16
14    callq  runCloudBasedSuperAI()
15    .loc   1 6 2                      # /tmp/compiler-
16    xorl  %eax, %eax
17    popq  %rcx
18    retq
19 .Lfunc_end0:
20    .size   main, .Lfunc_end0-main
```

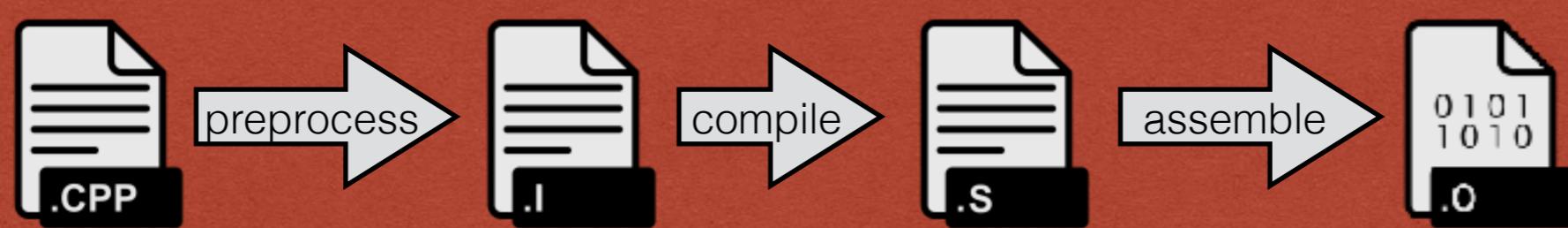








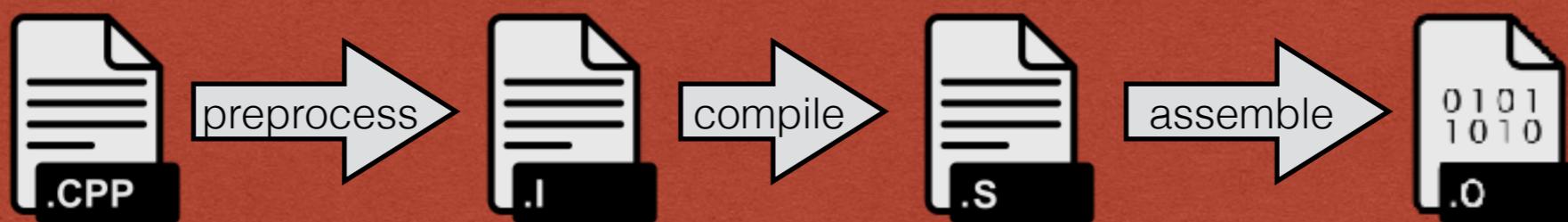




```
clang++ -c main.cpp -o main.o
```

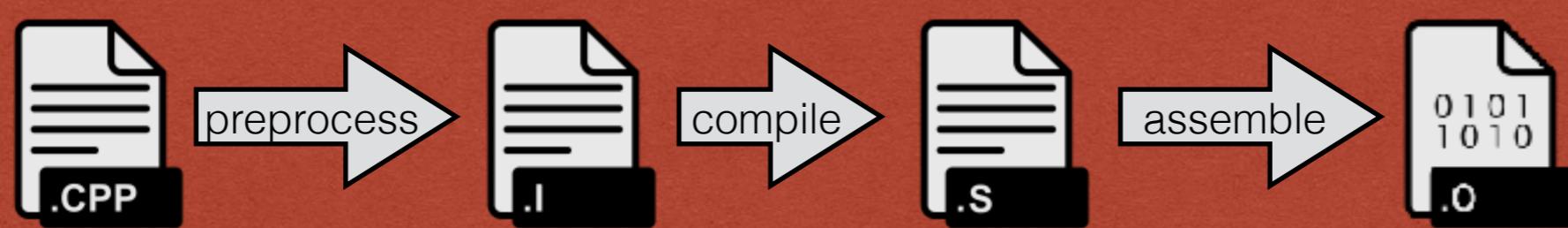


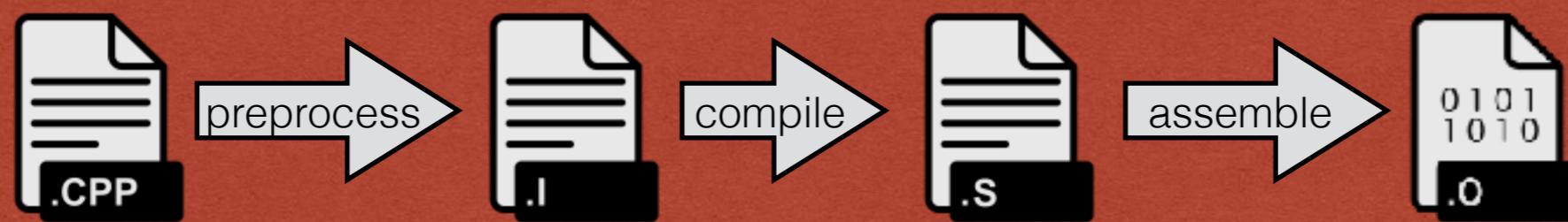
```
clang++ -c main.cpp -o main.o
```



```
clang++ -c main.cpp -o main.o
```







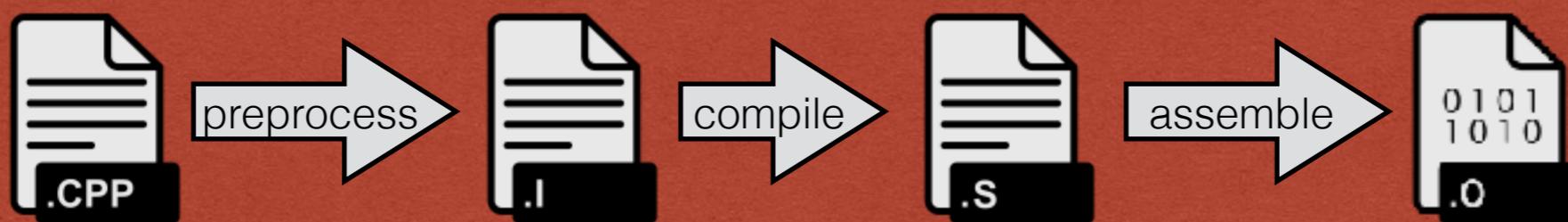
nm main.o



nm main.o

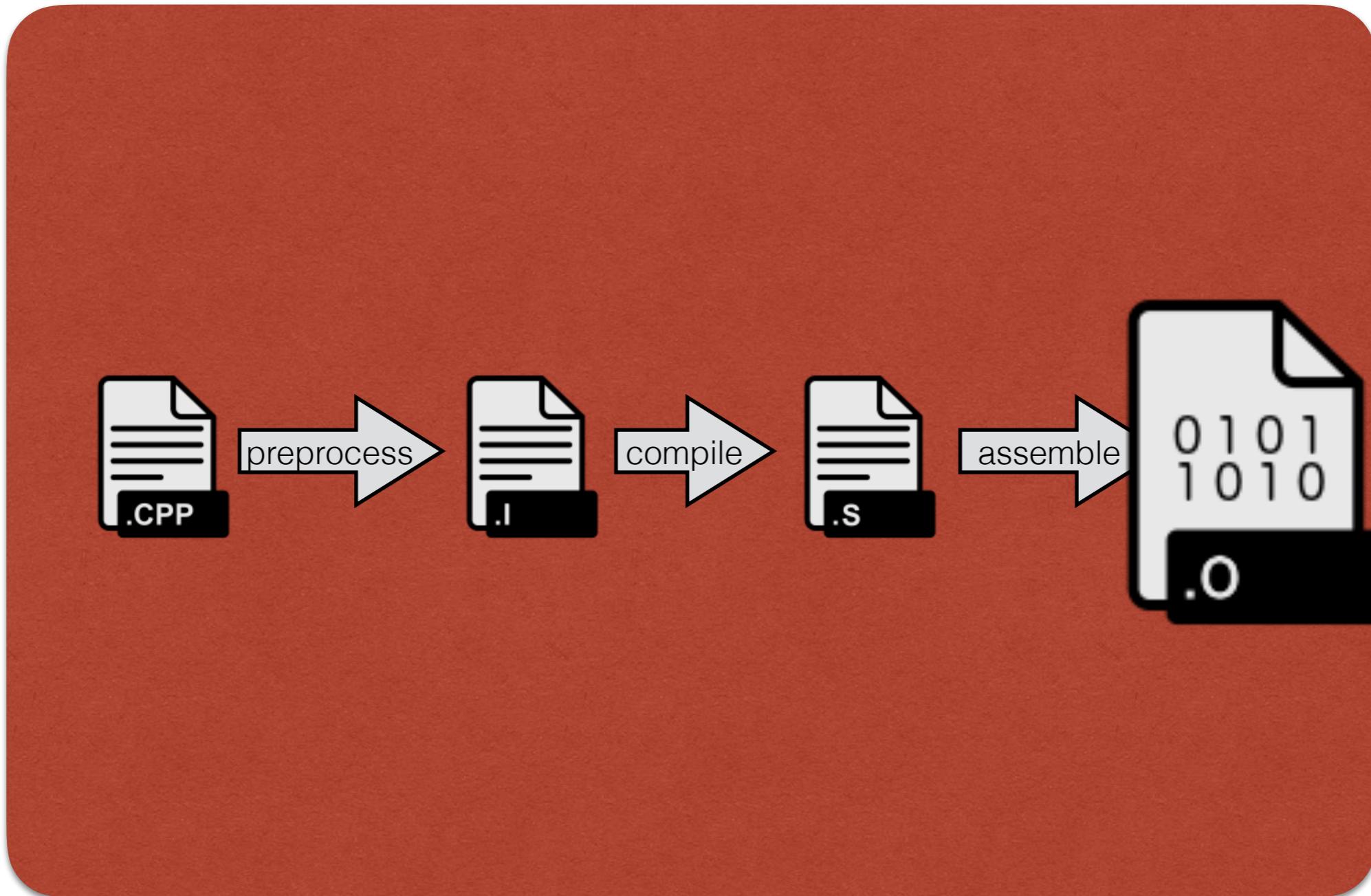


nm main.o



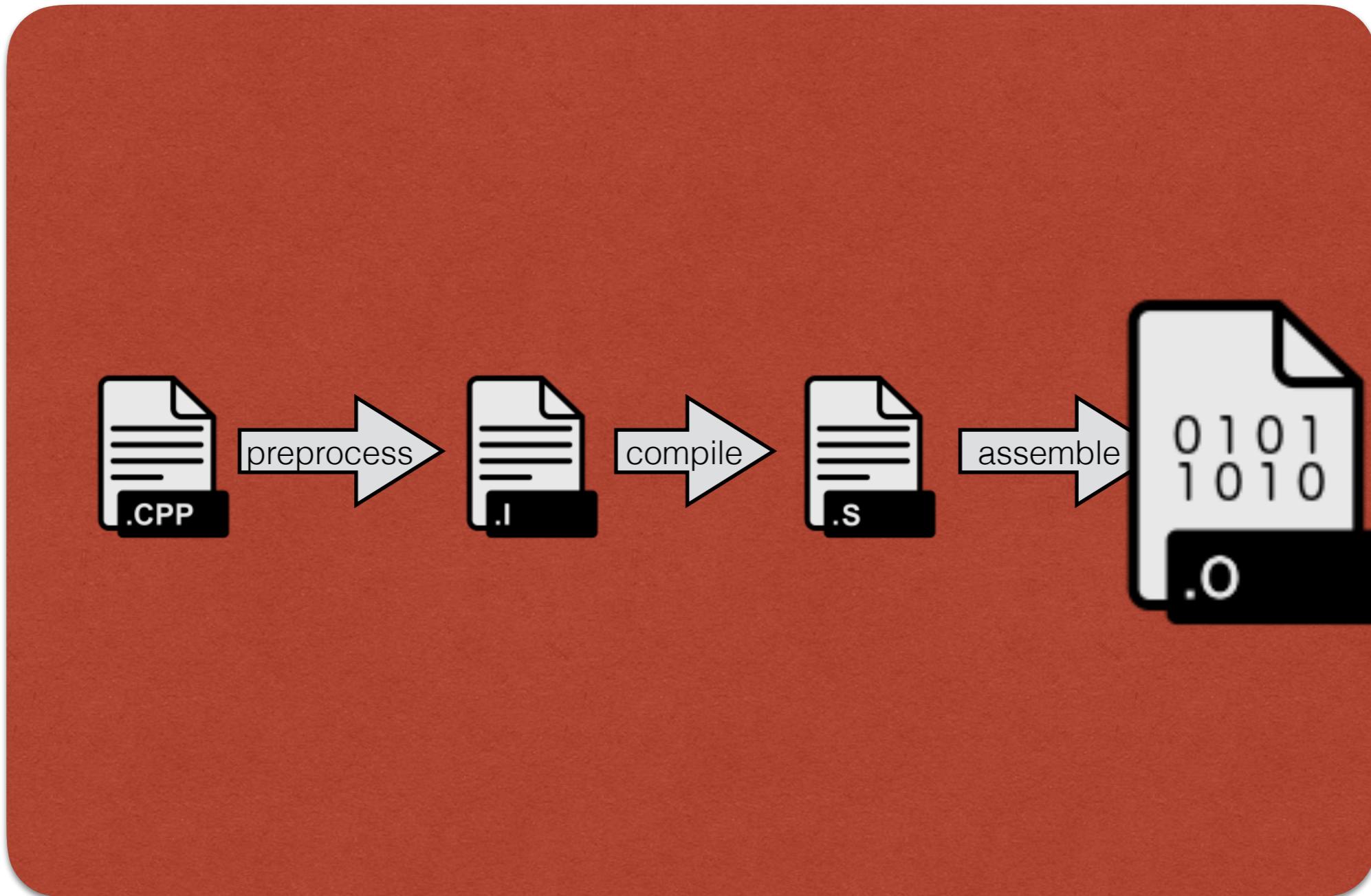
```
nm main.o
```

```
U __Z20runCloudBasedSuperAIv  
0000000000000000 T _main
```



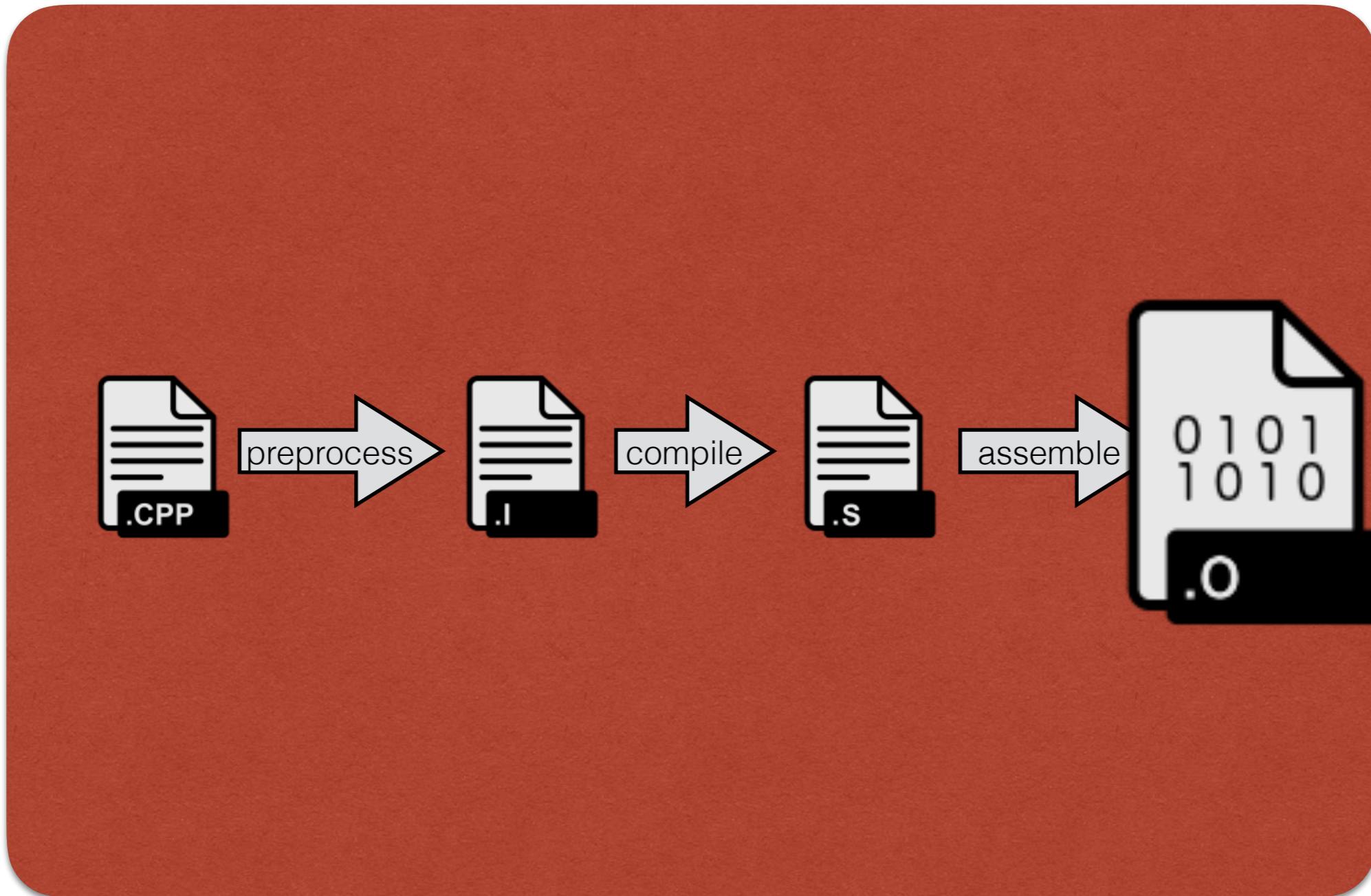
nm main.o

0000000000000000	U _Z20runCloudBasedSuperAIv
	T _main

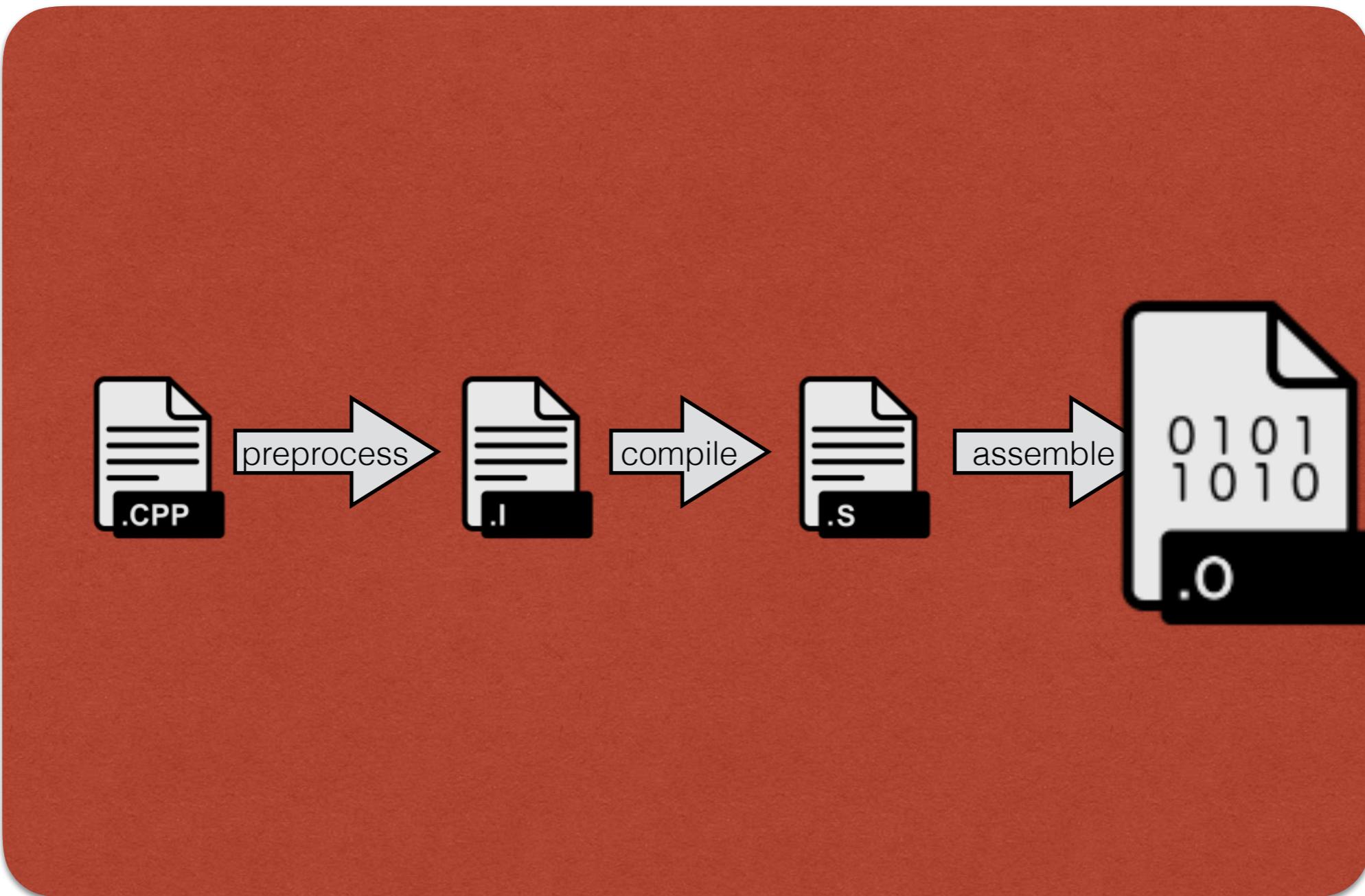


```
nm main.o
```

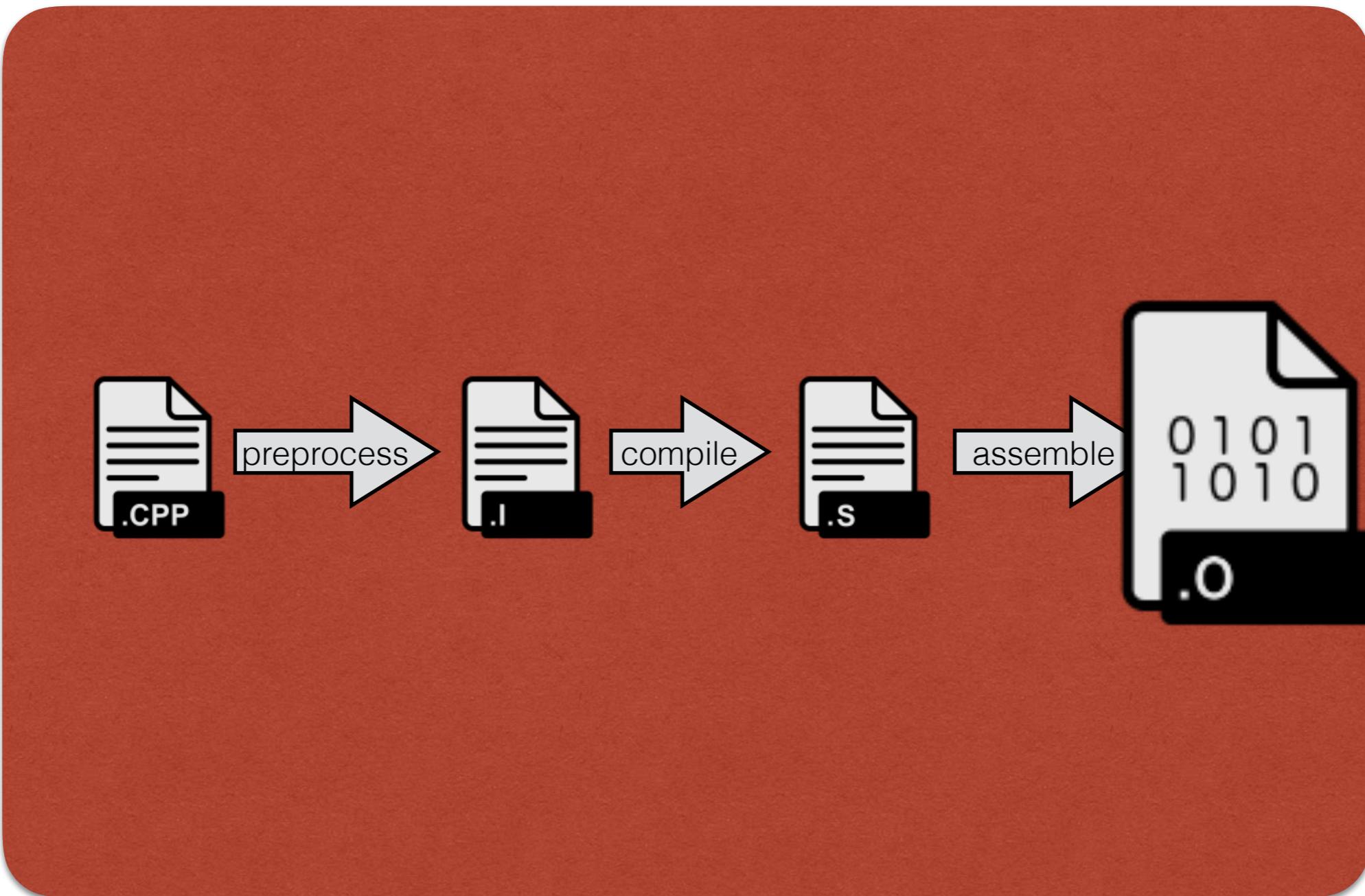
```
U __Z20runCloudBasedSuperAIv  
0000000000000000 T _main
```



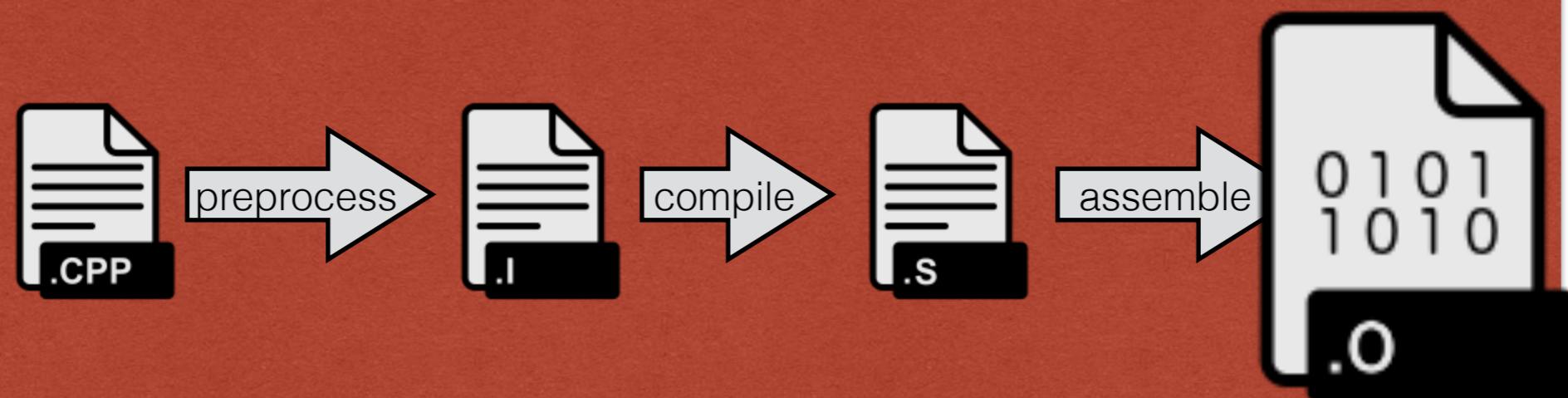
nm main.o



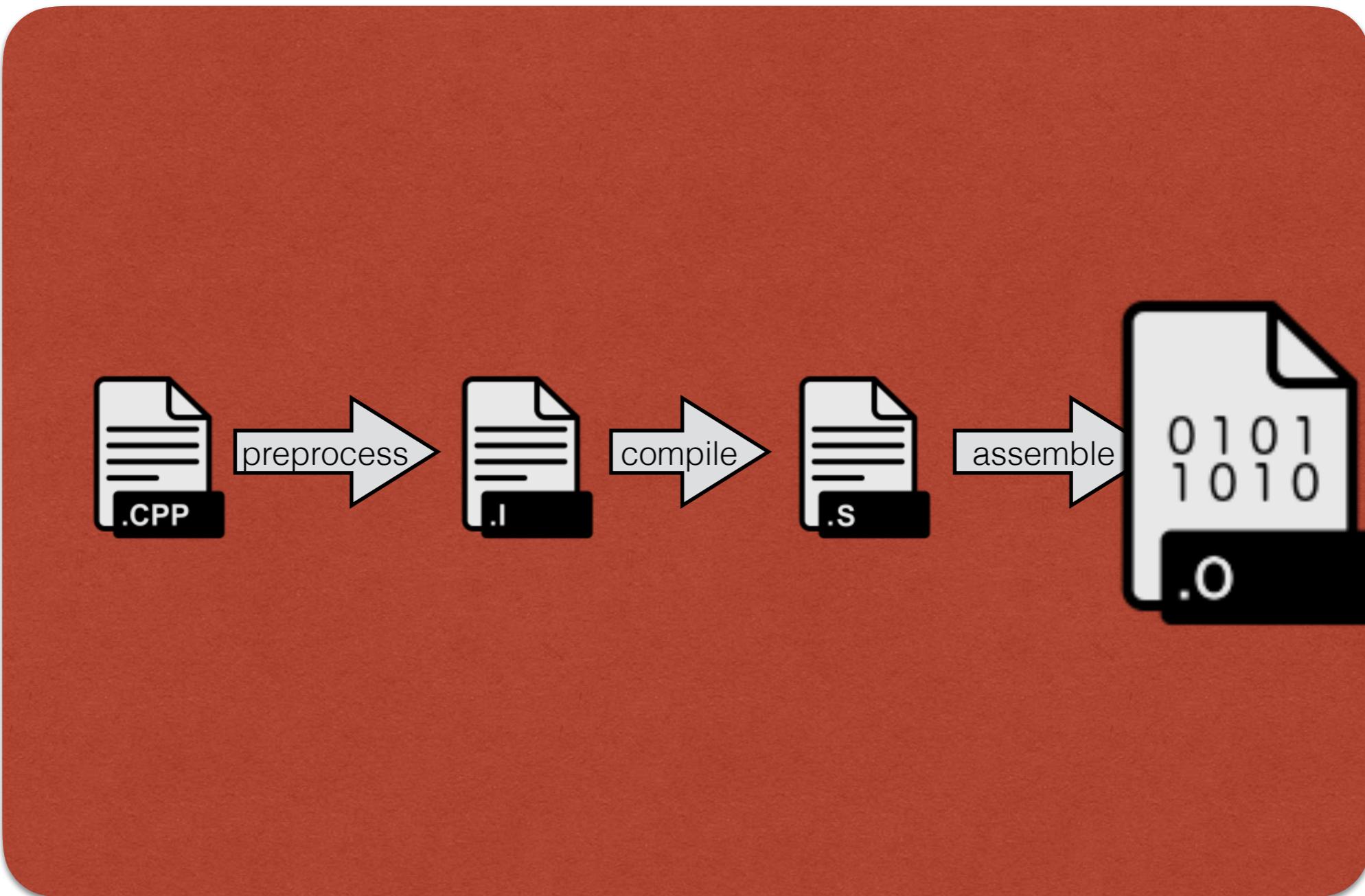
`otool -t main.o`



otool -t main.o



`otool -t main.o`

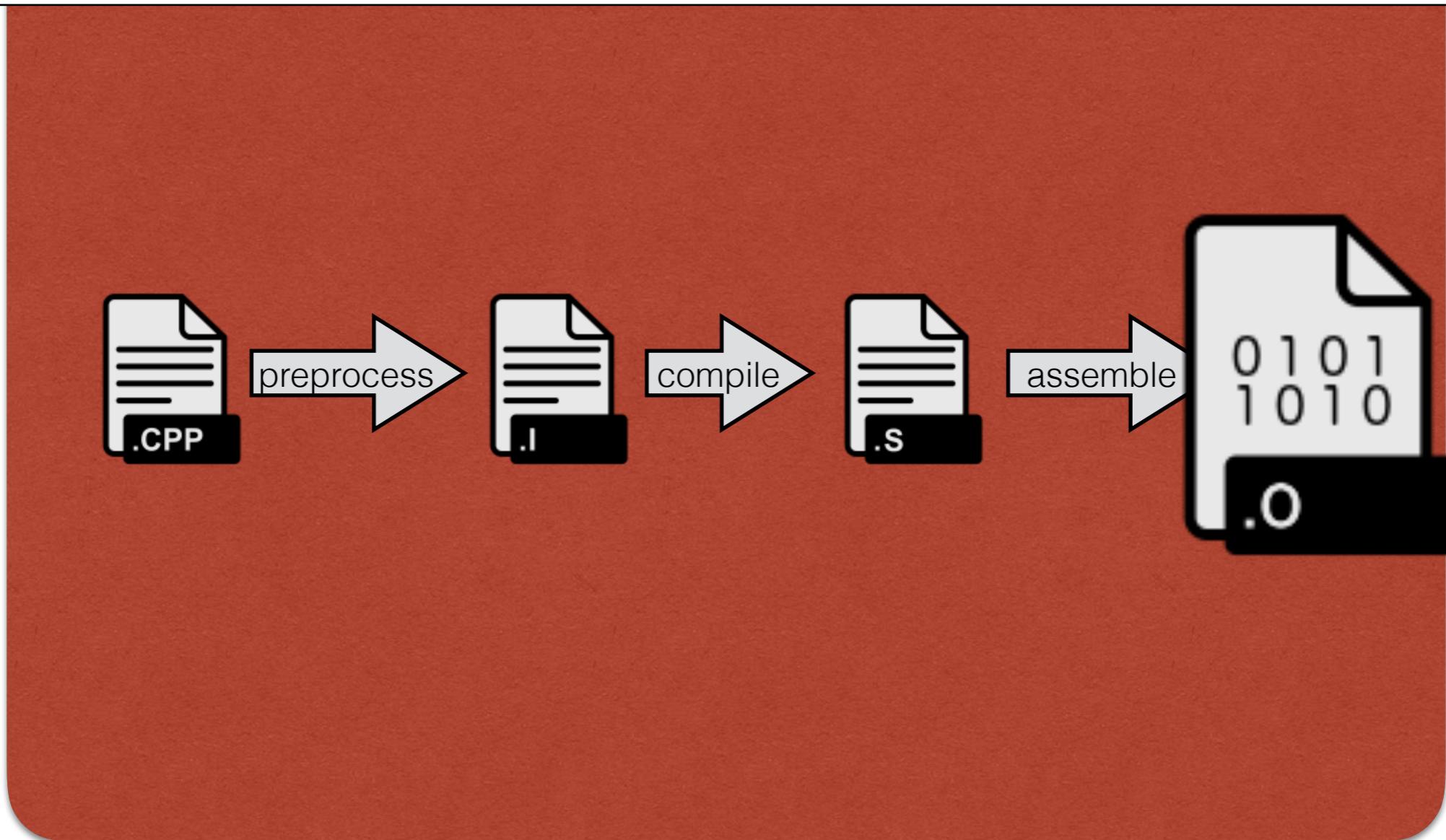


```
otool -t main.o
```

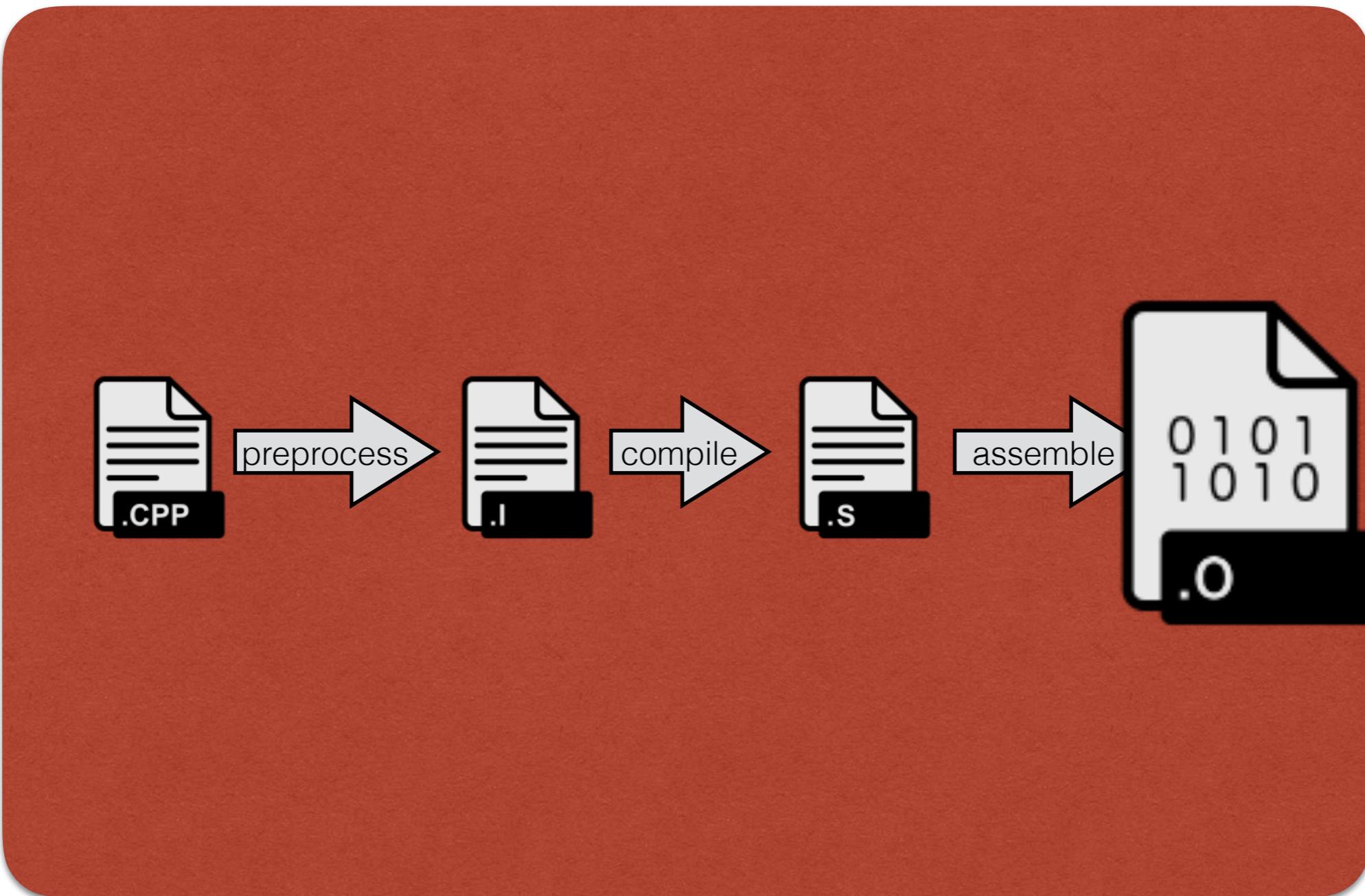
```
main.o:
```

```
Contents of (_TEXT,__text) section
```

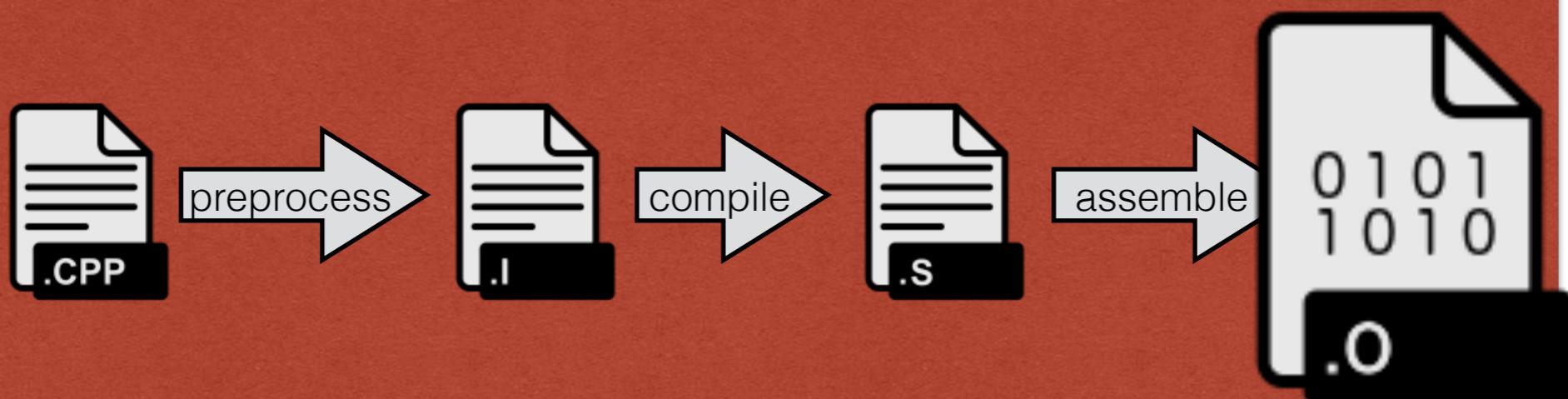
```
0000000000000000 55 48 89 e5 48 83 ec 10 c7 45 fc 00 00 00 00 e8  
0000000000000010 00 00 00 00 31 c0 48 83 c4 10 5d c3
```



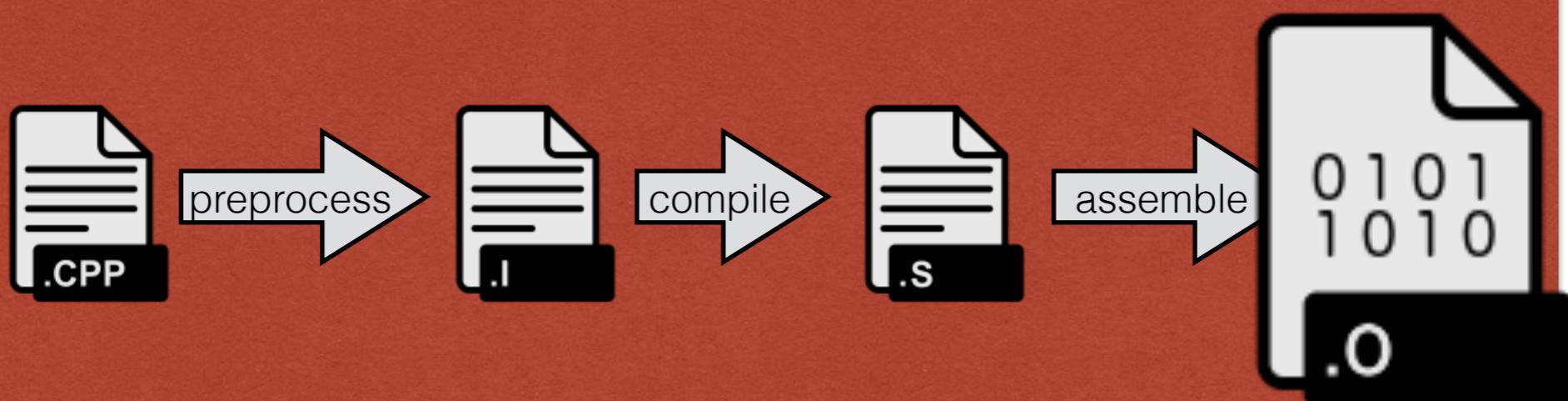
`otool -t main.o`



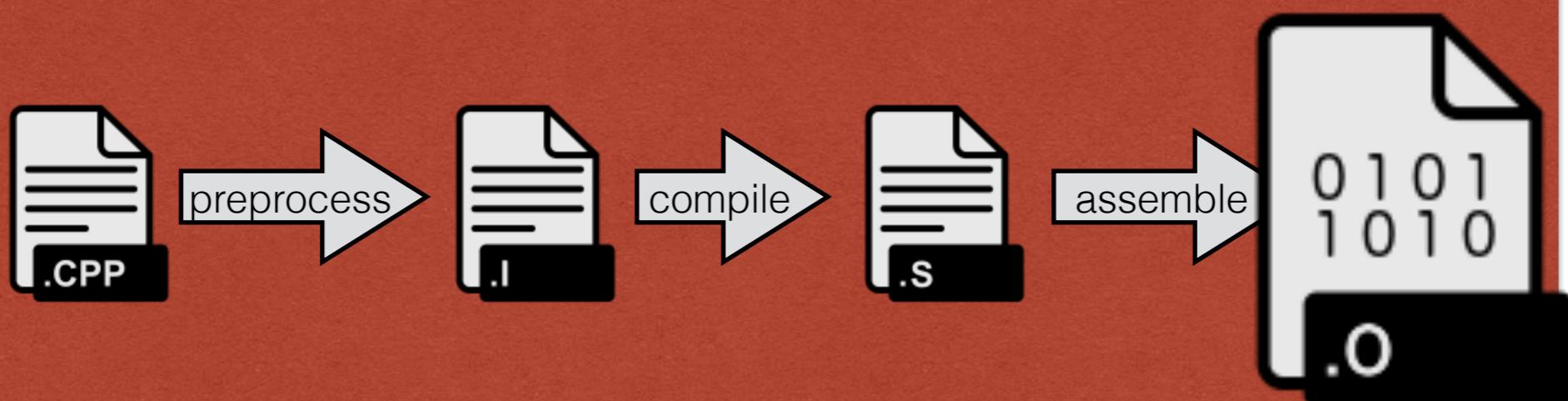
```
otool -t -v main.o
```



otool -t **-v** main.o



```
otool -t -v main.o
```



```
otool -t -v main.o
```

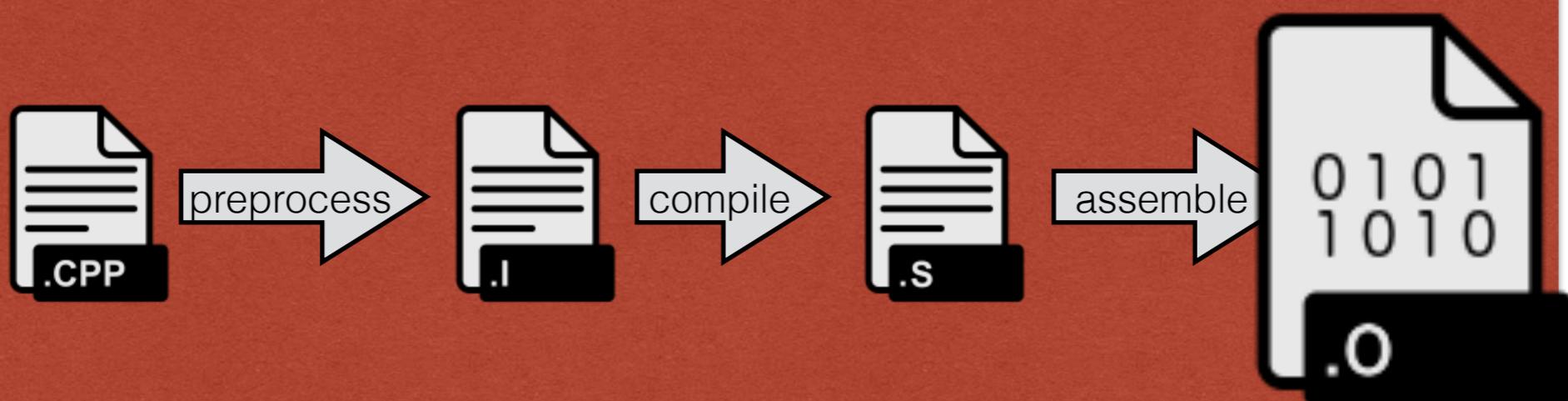
```
main.o:  
(__TEXT,__text) section
```

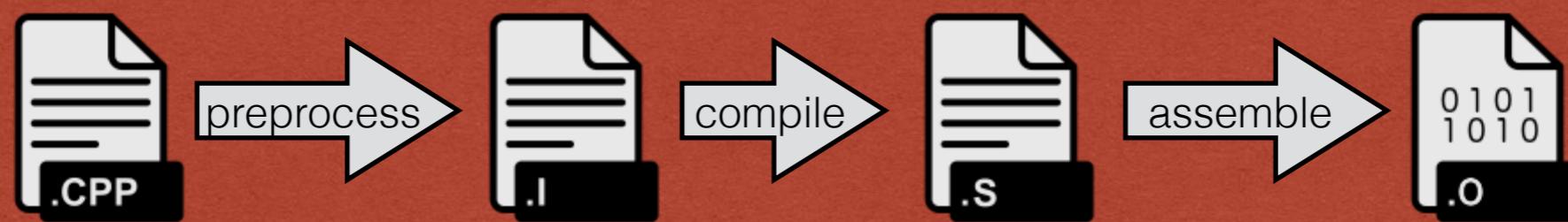
```
_main:
```

```
0000000000000000 pushq %rbp  
0000000000000001 movq %rsp, %rbp  
0000000000000004 subq $0x10, %rsp  
0000000000000008 movl $0x0, -0x4(%rbp)  
000000000000000f callq 0x14  
0000000000000014 xorl %eax, %eax  
0000000000000016 addq $0x10, %rsp  
000000000000001a popq %rbp  
000000000000001b retq
```



```
otool -t -v main.o
```



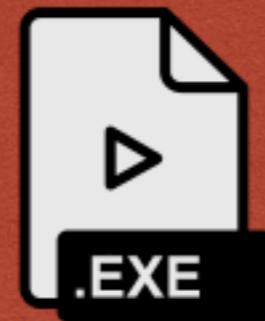


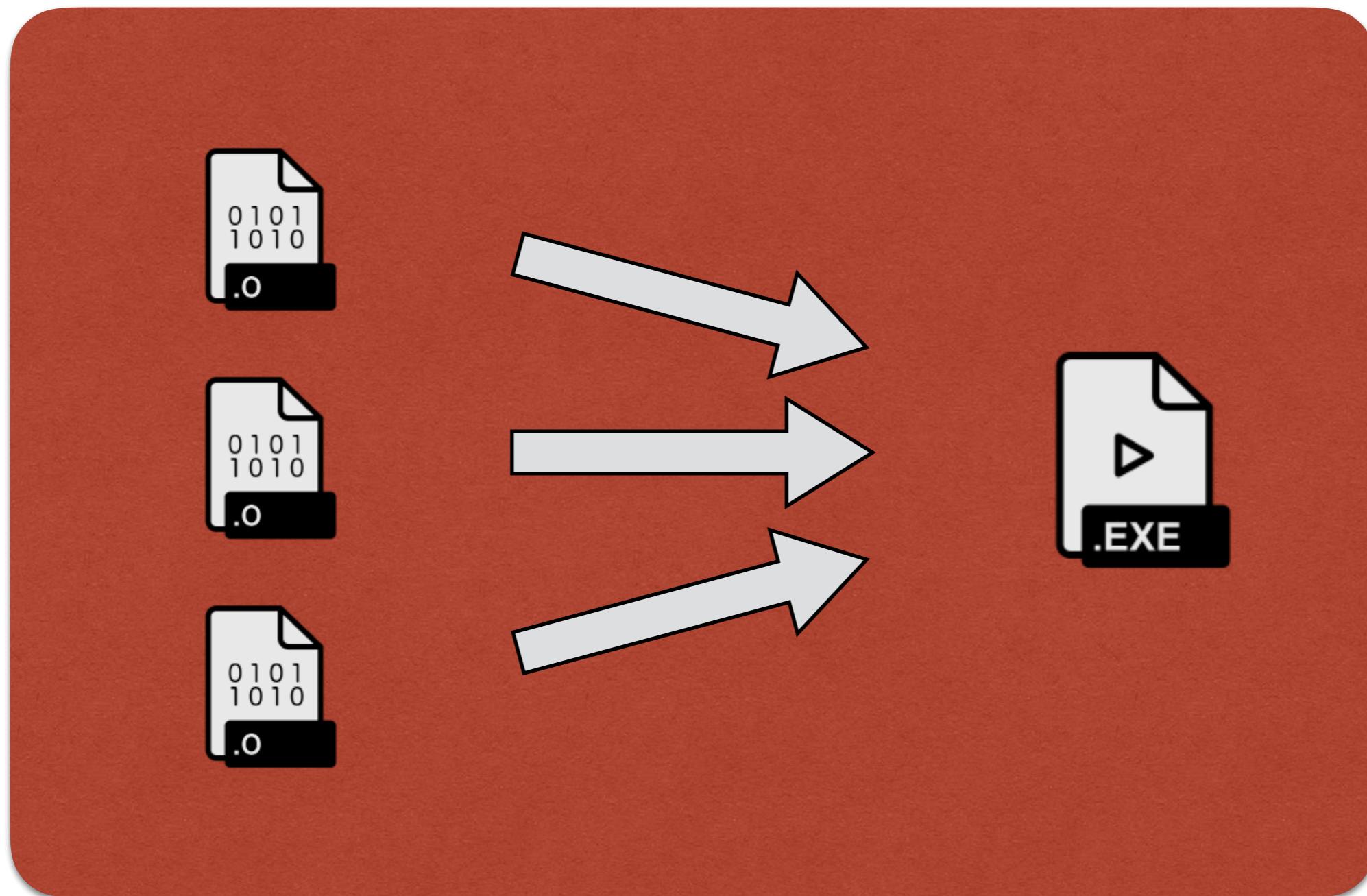


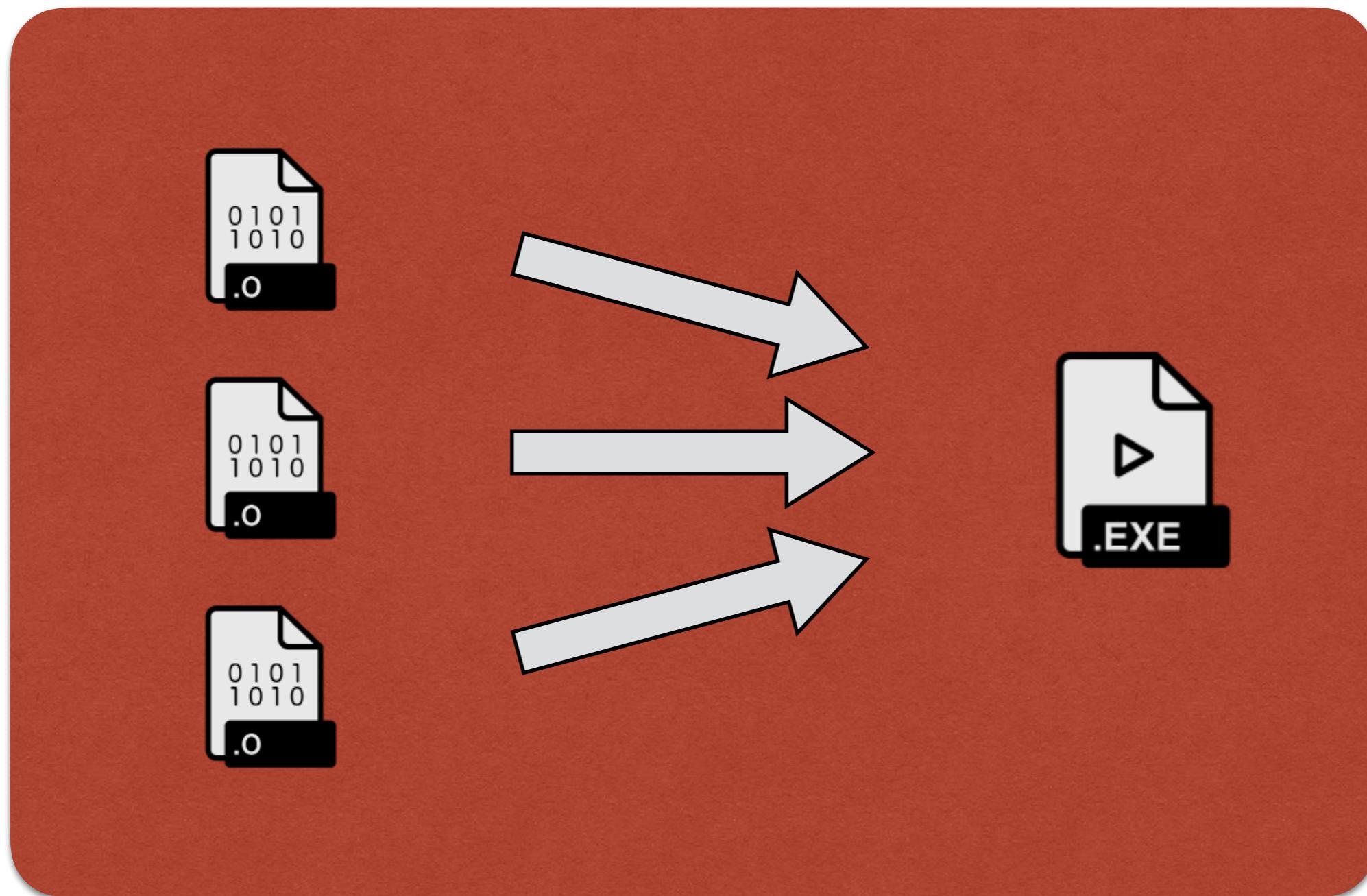




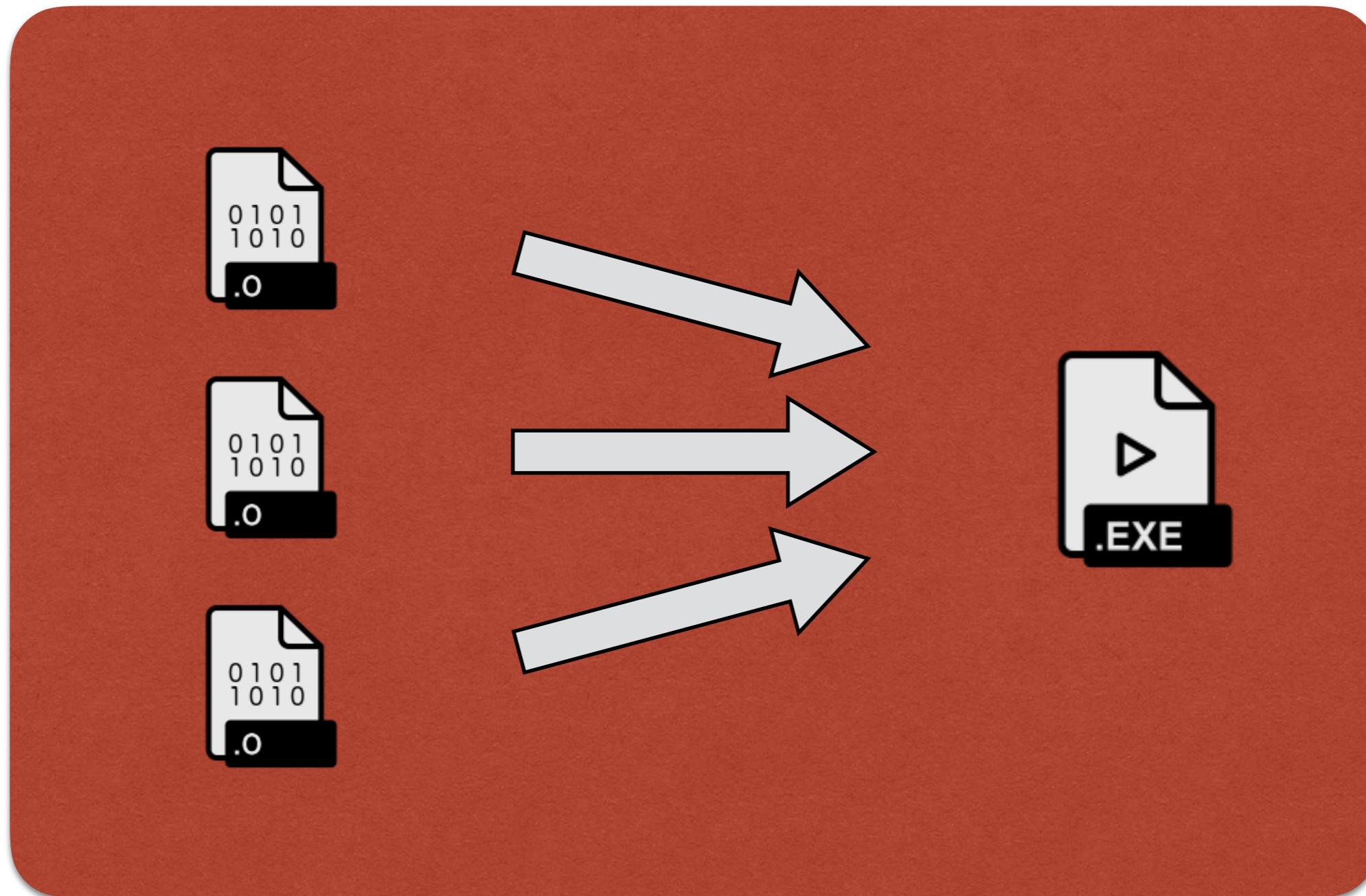








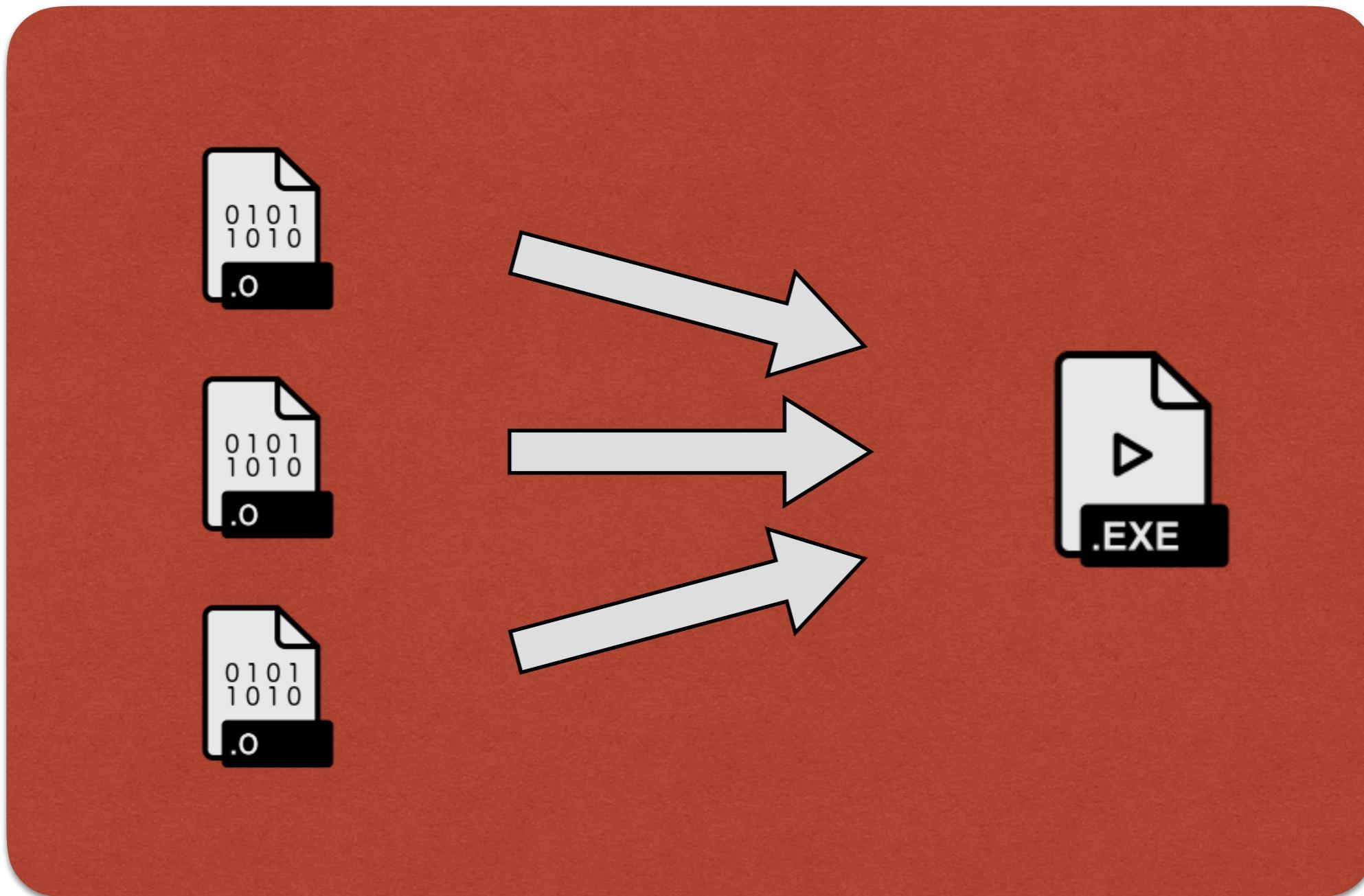
Linkage



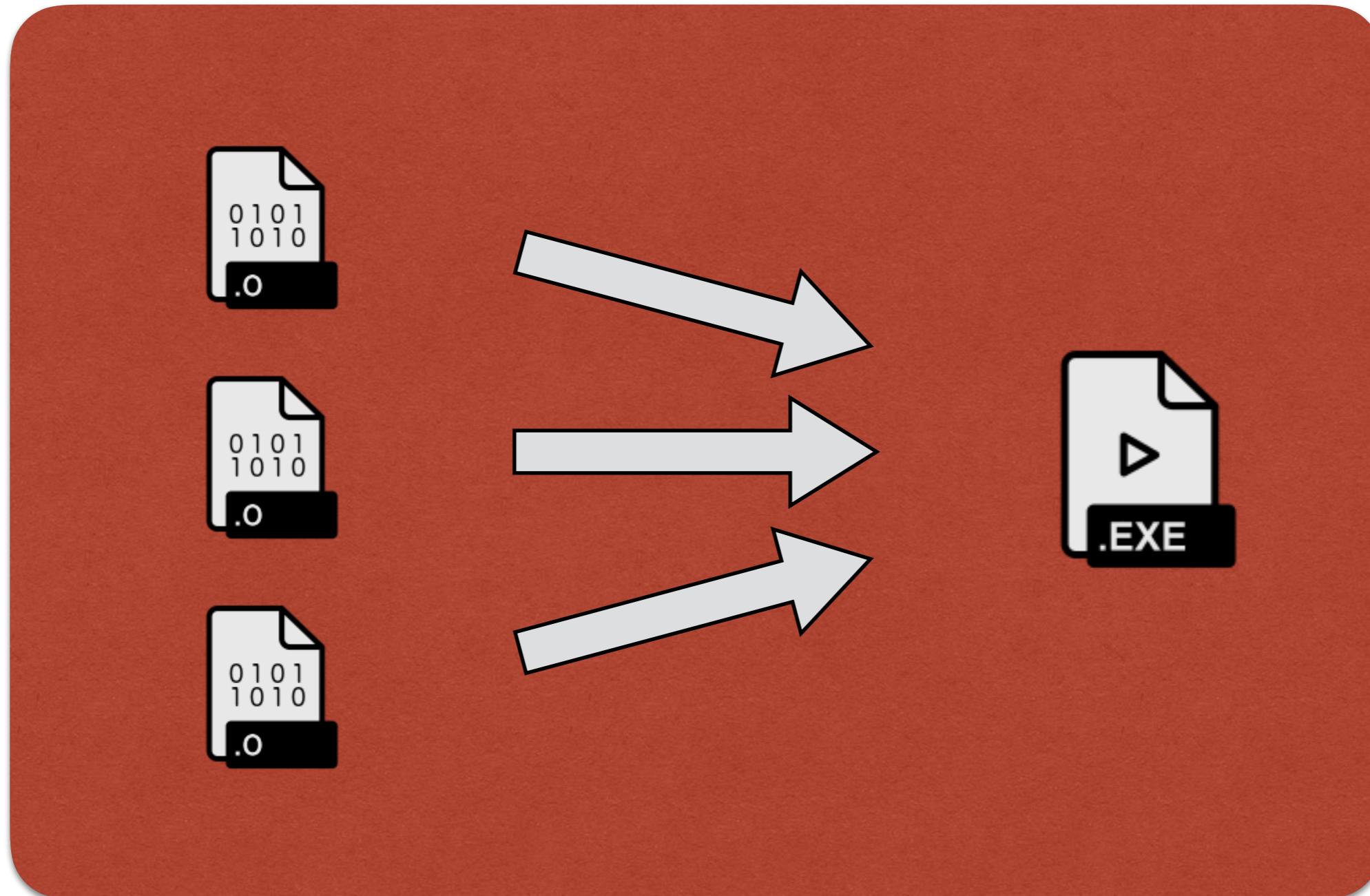
Linkage

```
1 // Global namespace
2 int i;                                // External linkage (implicit)
3 const int ci;                            // Internal linkage (implicit)
4 extern const int eci;                   // External linkage (explicit)
5 static int si;                            // Internal linkage (explicit)
6
7 void f();                               // External linkage (implicit)
8 static void g();                          // Internal linkage (explicit)
9
10 // Unnamed namespace
11 namespace {
12     int ni;                             // Internal linkage
13     const int nci;                      // Internal linkage
14     extern const int neci;              // Internal linkage (extern is ignored)
15     static int nsi;                     // Internal linkage (static is ignored)
16
17     void nf();                           // Internal linkage
18     static void ng();                   // Internal linkage (static is ignored)
19 }
```

Linkage



Linkage (2)



Linkage (2)

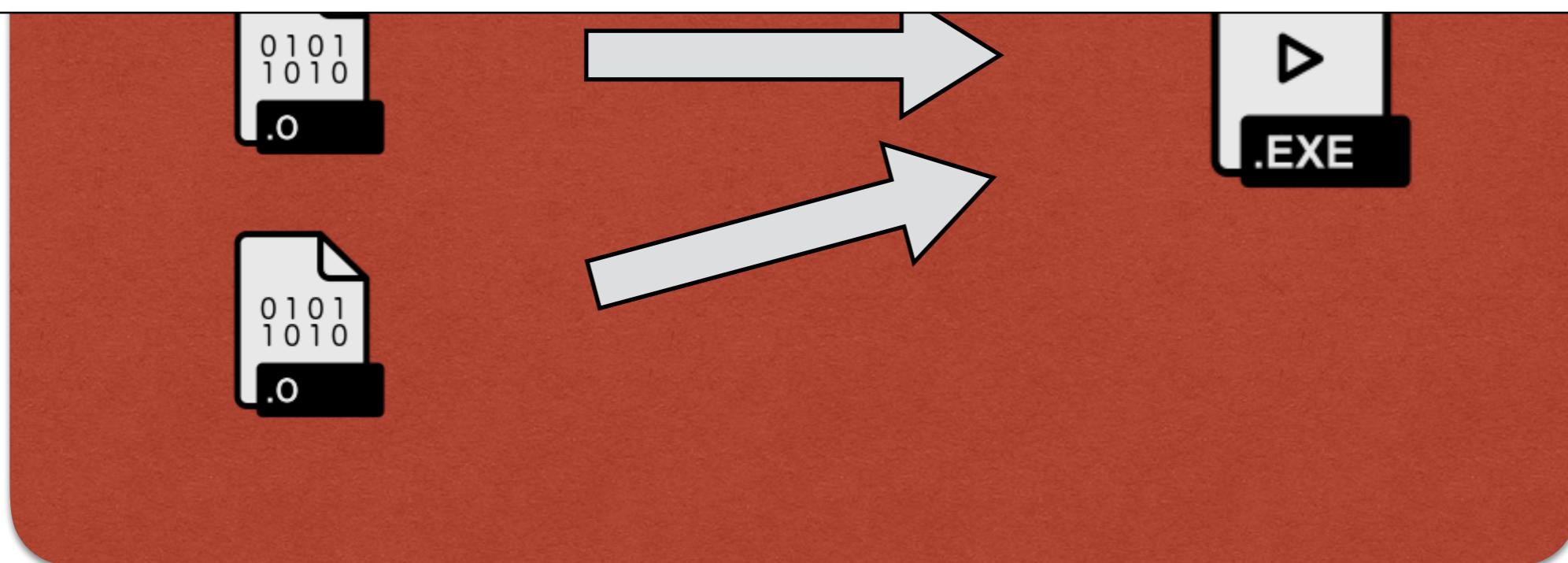
Explicit:

```
extern  
static  
namespace {...}
```

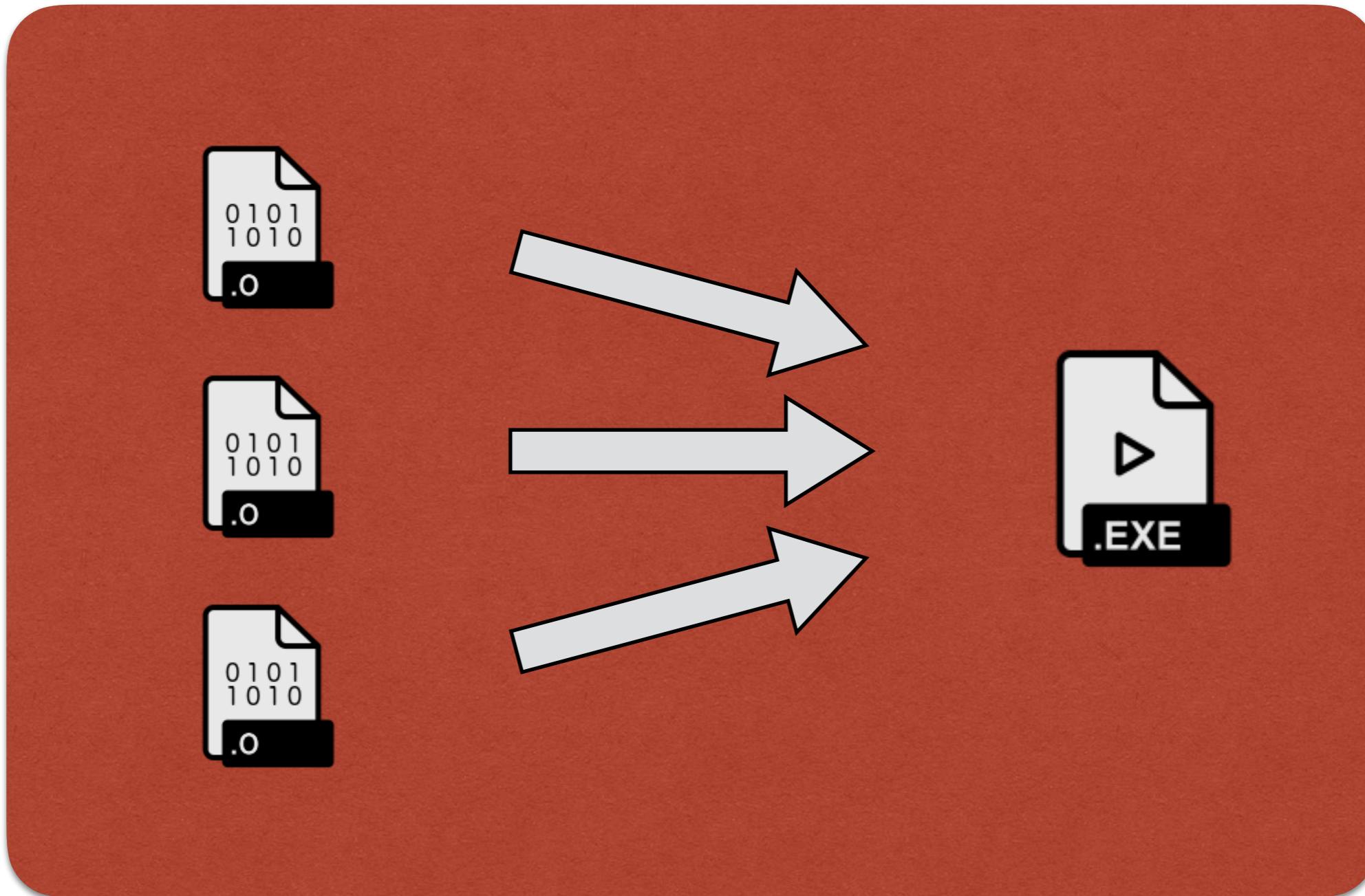
Implicit:

```
const int => Internal  
int       => External
```

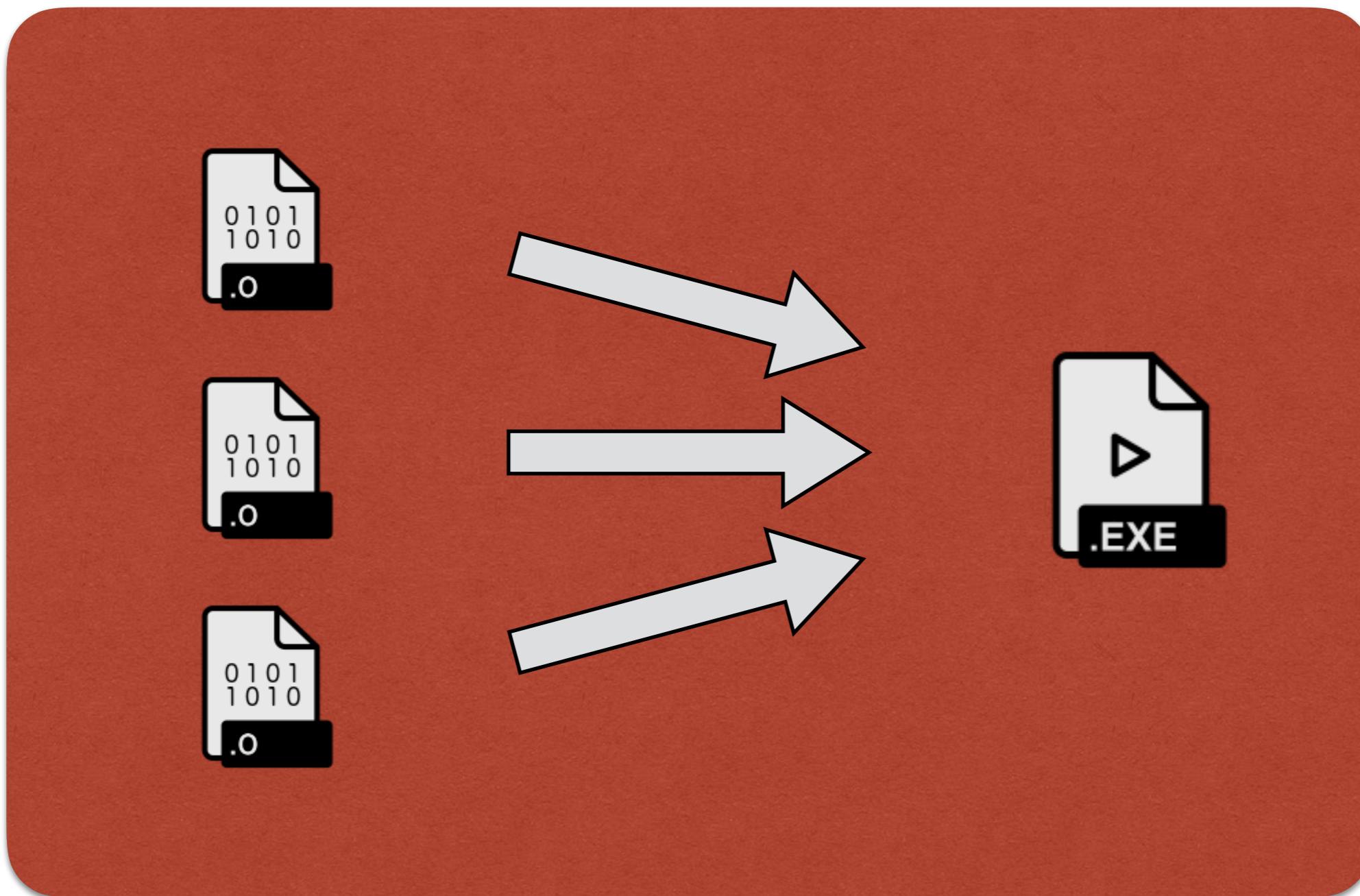
```
void f()  => External
```



Linkage (2)



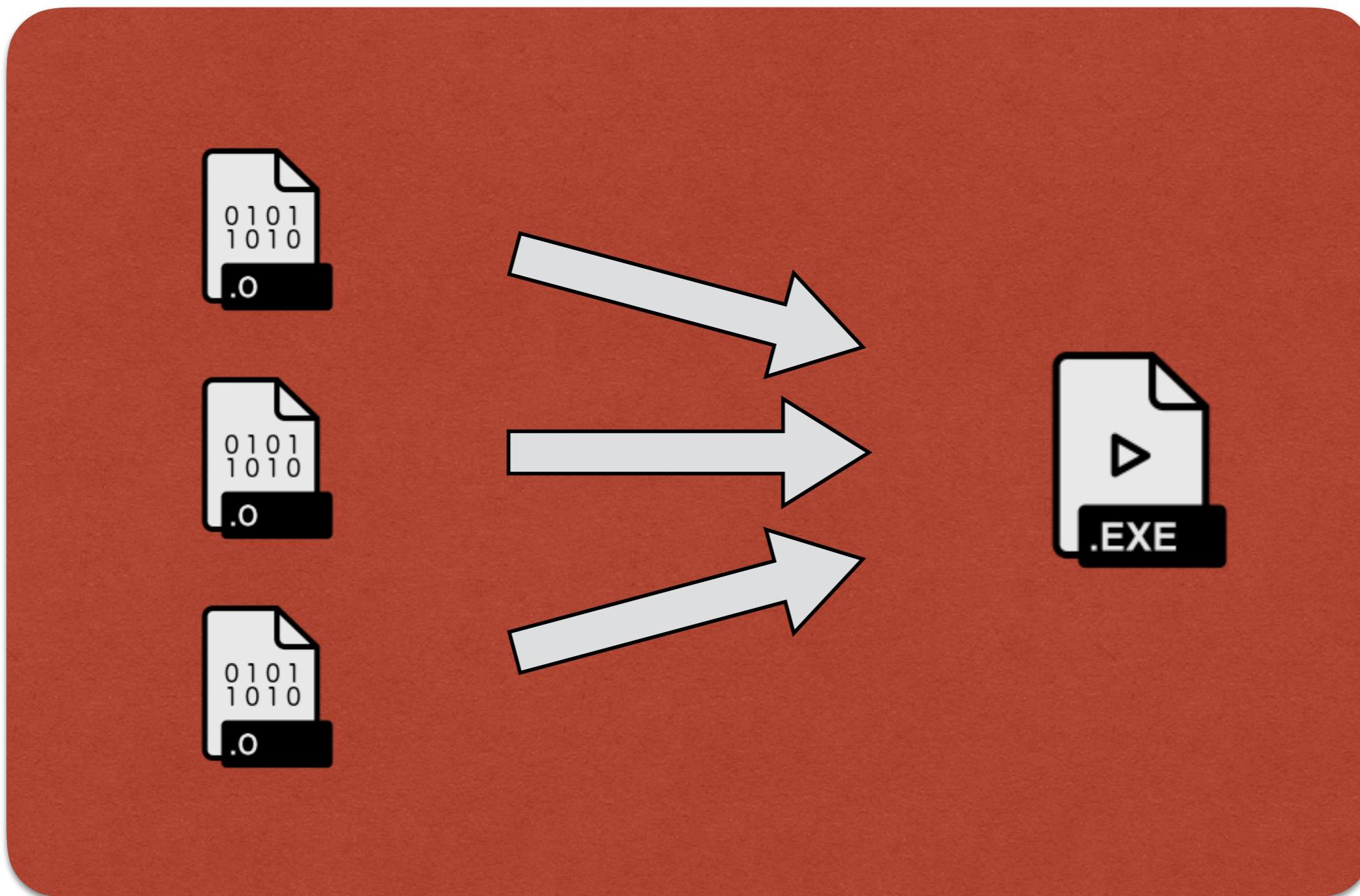
Name mangling



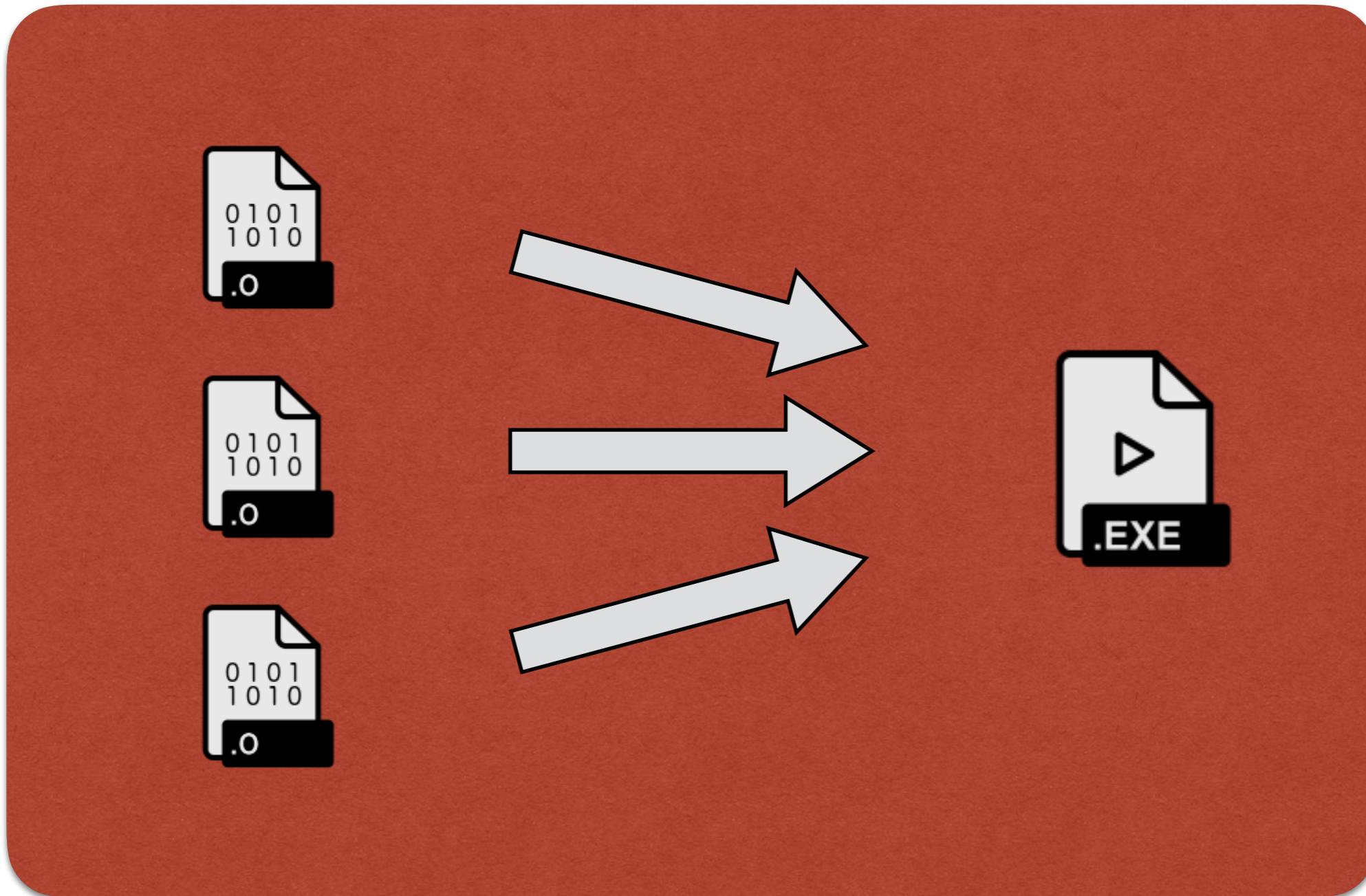
Name mangling

```
1 // Global namespace
2 int i;                      // _i
3 const int ci;                // __ZL2ci
4 extern const int eci;         // __eci
5 static int si;                // __ZL2si
6
7 void f();                    // __Z1fv
8 static void g();              // __ZL1gv
9
10 // Unnamed namespace
11 namespace {
12     int ni;                  // __ZN12_GLOBAL__N_12niE
13     const int nci;            // __ZN12_GLOBAL__N_1L3nciE
14     extern const int neci;    // __ZN12_GLOBAL__N_1L3neciE
15     static int nsi;            // __ZN12_GLOBAL__N_1L3nsiE
16
17     void nf();                // __ZN12_GLOBAL__N_12nfEv
18     static void ng();          // __ZN12_GLOBAL__N_1L2ngEv
19 }
```

Name mangling



Name mangling (2)



Name mangling (2)

```
1 // Function overloading
2 int max(const int&, const int&);
3 float max(const float&, const float&);
4 double max(const double&, const double&);
5
6 // Templates
7 template <typename T>
8 void max(const T&, const T&);
9
10 int main()
11 {
12     max(1, 1);           // max(int, int);          __Z3maxRKiS0_
13     max(1.f, 1.f);       // max(float, float);      __Z3maxRKfS0_
14     max(1., 1.);         // max(double, double);    __Z3maxRKdS0_
15     max<>(1, 1);        // max<int>(...);          __Z3maxIiET_RKS0_S2_
16     max<>(1.f, 1.f);     // max<float>(...);        __Z3maxIfET_RKS0_S2_
17     max<>(1., 1.);       // max<double>(...);       __Z3maxIdET_RKS0_S2_
18 }
```

Name mangling (2)

```
1 // Function overloading
2 int max(const int&, const int&);
3 float max(const float&, const float&);
4 double max(const double&, const double&);
5
6 // Templates
7 template <typename T>
8 void max(const T&, const T&);
9
10 int main()
11 {
12     max(1, 1);           // max(int, int);          __Z3maxRKiS0_
13     max(1.f, 1.f);       // max(float, float);      __Z3maxRKfS0_
14     max(1., 1.);         // max(double, double);    __Z3maxRKdS0_
15     max<>(1, 1);        // max<int>(...);          __Z3maxIiET_RKS0_S2_
16     max<>(1.f, 1.f);     // max<float>(...);        __Z3maxIfET_RKS0_S2_
17     max<>(1., 1.);       // max<double>(...);       __Z3maxIdET_RKS0_S2_
18 }
```

Name mangling (2)

```
1 // Function overloading
2 int max(const int&, const int&);
3 float max(const float&, const float&);
4 double max(const double&, const double&);
5
6 // Templates
7 template <typename T>
8 void max(const T&, const T&);
9
10 int main()
11 {
12     max(1, 1);           // max(int, int);          __Z3maxRKiS0_
13     max(1.f, 1.f);       // max(float, float);      __Z3maxRKfS0_
14     max(1., 1.);         // max(double, double);    __Z3maxRKdS0_
15     max<>(1, 1);        // max<int>(...);          __Z3maxIiET_RKS0_S2_
16     max<>(1.f, 1.f);     // max<float>(...);        __Z3maxIfET_RKS0_S2_
17     max<>(1., 1.);       // max<double>(...);       __Z3maxIdET_RKS0_S2_
18 }
```

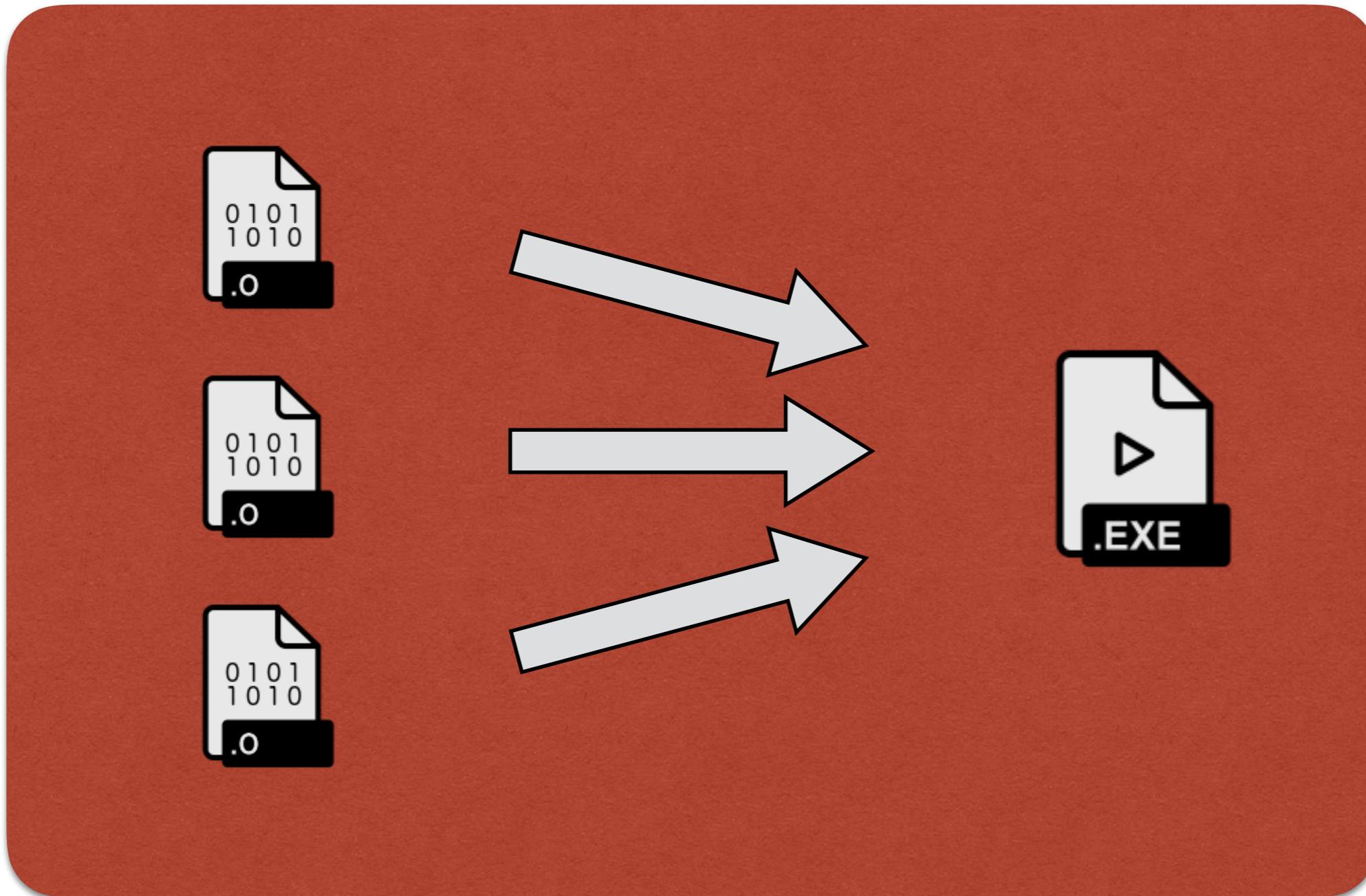
Name mangling (2)

```
1 // Function overloading
2 int max(const int&, const int&);
3 float max(const float&, const float&);
4 double max(const double&, const double&);
5
6 // Templates
7 template <typename T>
8 void max(const T&, const T&);
9
10 int main()
11 {
12     max(1, 1);           // max(int, int);          __Z3maxRKiS0_
13     max(1.f, 1.f);       // max(float, float);      __Z3maxRKfS0_
14     max(1., 1.);         // max(double, double);    __Z3maxRKdS0_
15     max<>(1, 1);        // max<int>(...);          __Z3maxXiET_RKS0_S2_
16     max<>(1.f, 1.f);     // max<float>(...);        __Z3maxXfET_RKS0_S2_
17     max<>(1., 1.);       // max<double>(...);       __Z3maxXdET_RKS0_S2_
18 }
```

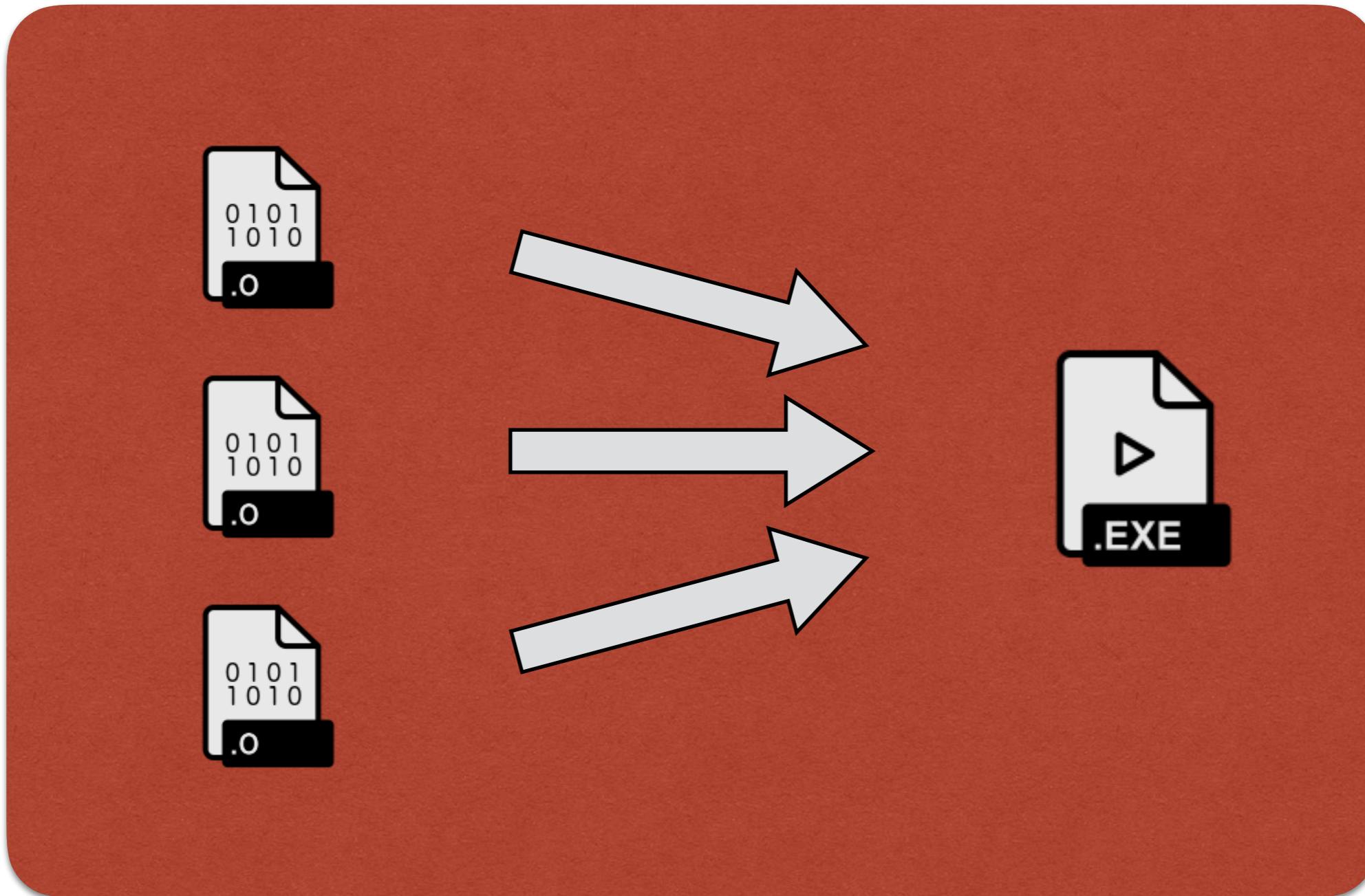
Name mangling (2)

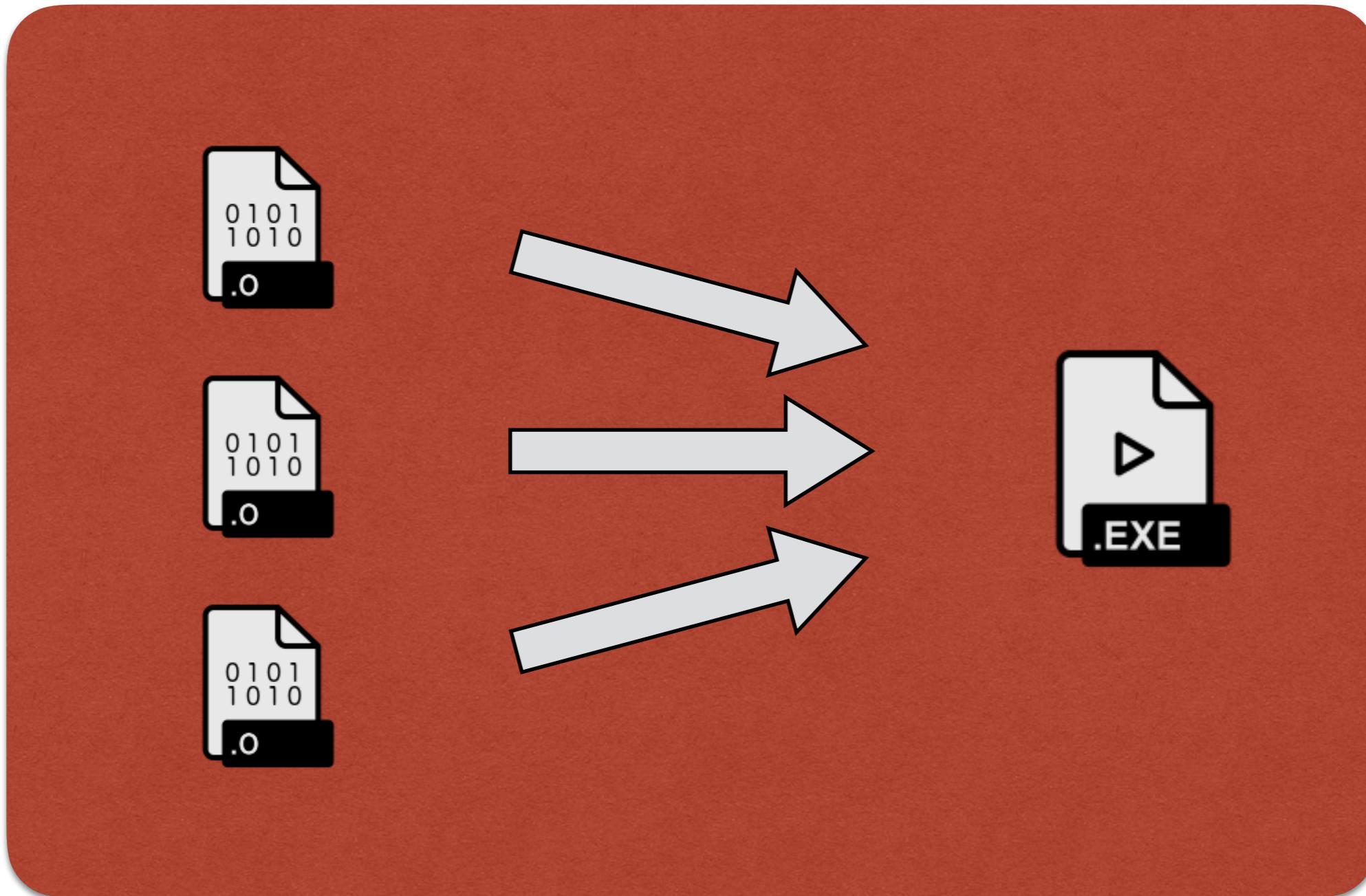
```
1 // Function overloading
2 int max(const int&, const int&);
3 float max(const float&, const float&);
4 double max(const double&, const double&);
5
6 // Templates
7 template <typename T>
8 void max(const T&, const T&);
9
10 int main()
11 {
12     max(1, 1);           // max(int, int);          __Z3maxRKiS0_
13     max(1.f, 1.f);       // max(float, float);      __Z3maxRKfS0_
14     max(1., 1.);         // max(double, double);    __Z3maxRKdS0_
15     max<>(1, 1);        // max<int>(...);          __Z3maxIiET_RKS0_S2_
16     max<>(1.f, 1.f);     // max<float>(...);        __Z3maxIfET_RKS0_S2_
17     max<>(1., 1.);       // max<double>(...);       __Z3maxIdET_RKS0_S2_
18 }
```

Name mangling (2)

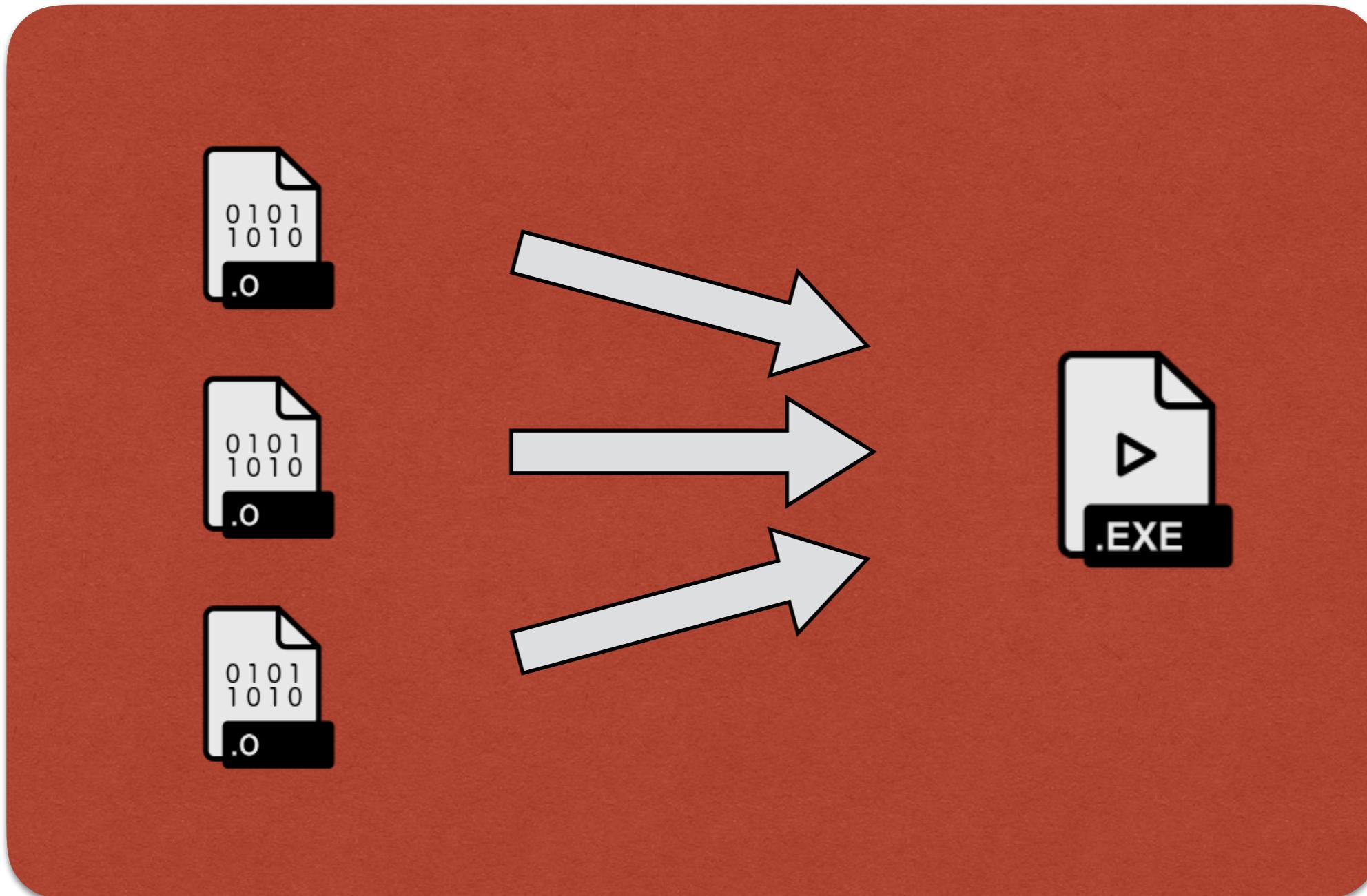


c++filt __Z3maxIfET_RKS0_S2_



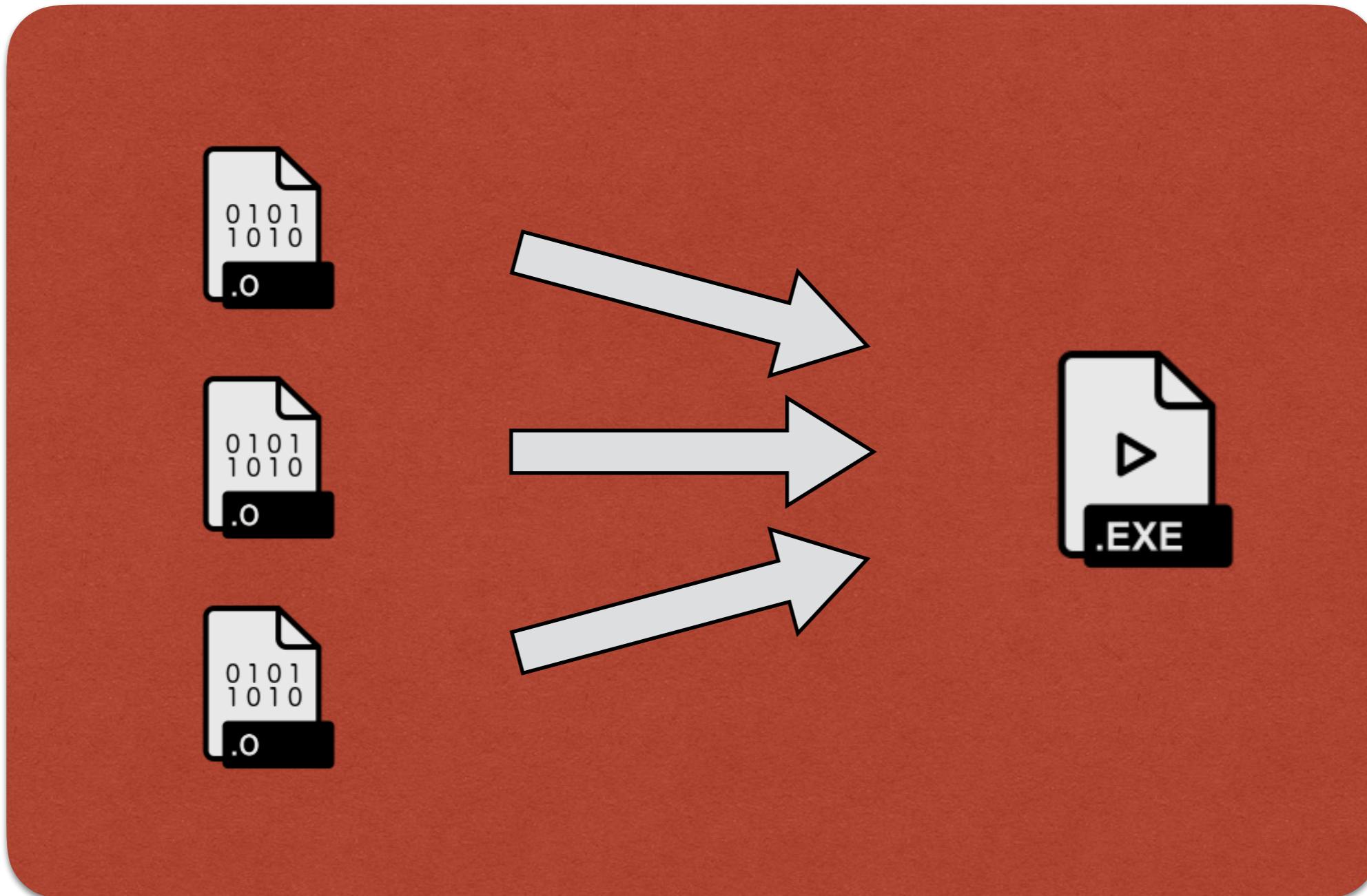


c++filt __Z3maxIfET_RKS0_S2_

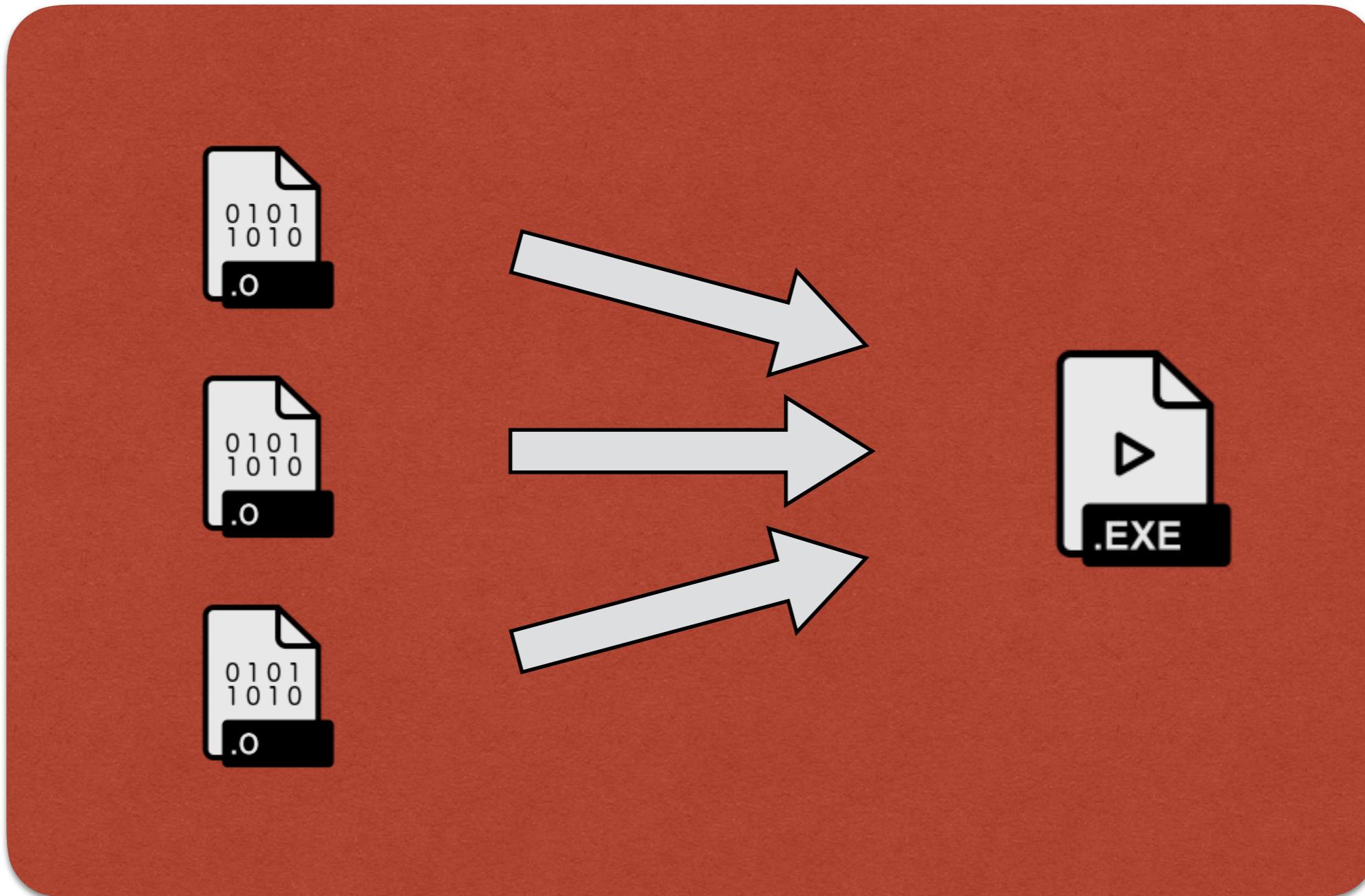


c++filt __Z3maxIfET_RKS0_S2_

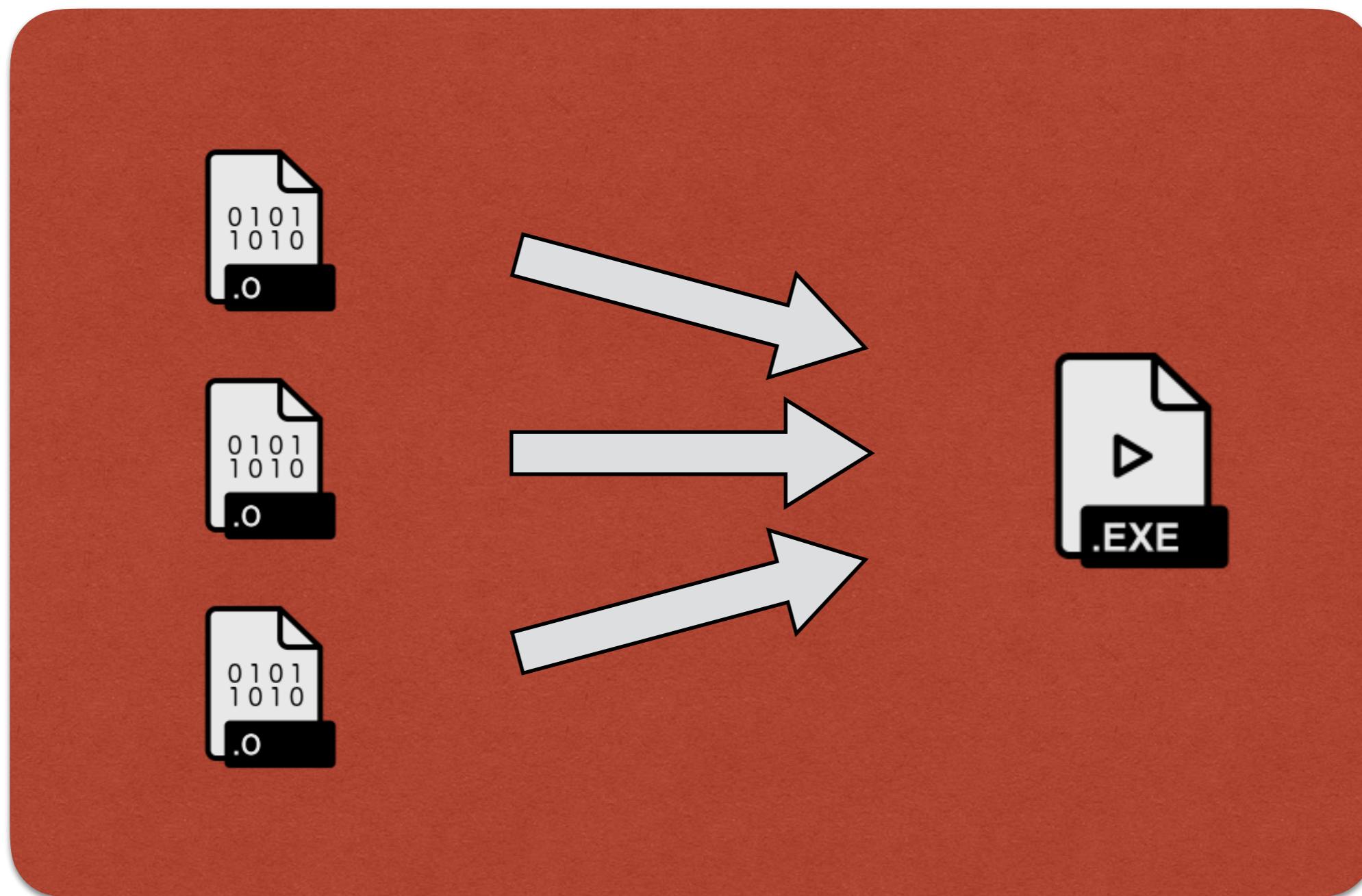
```
float max<float>(float const&, float const&)
```



c++filt __Z3maxIfET_RKS0_S2_

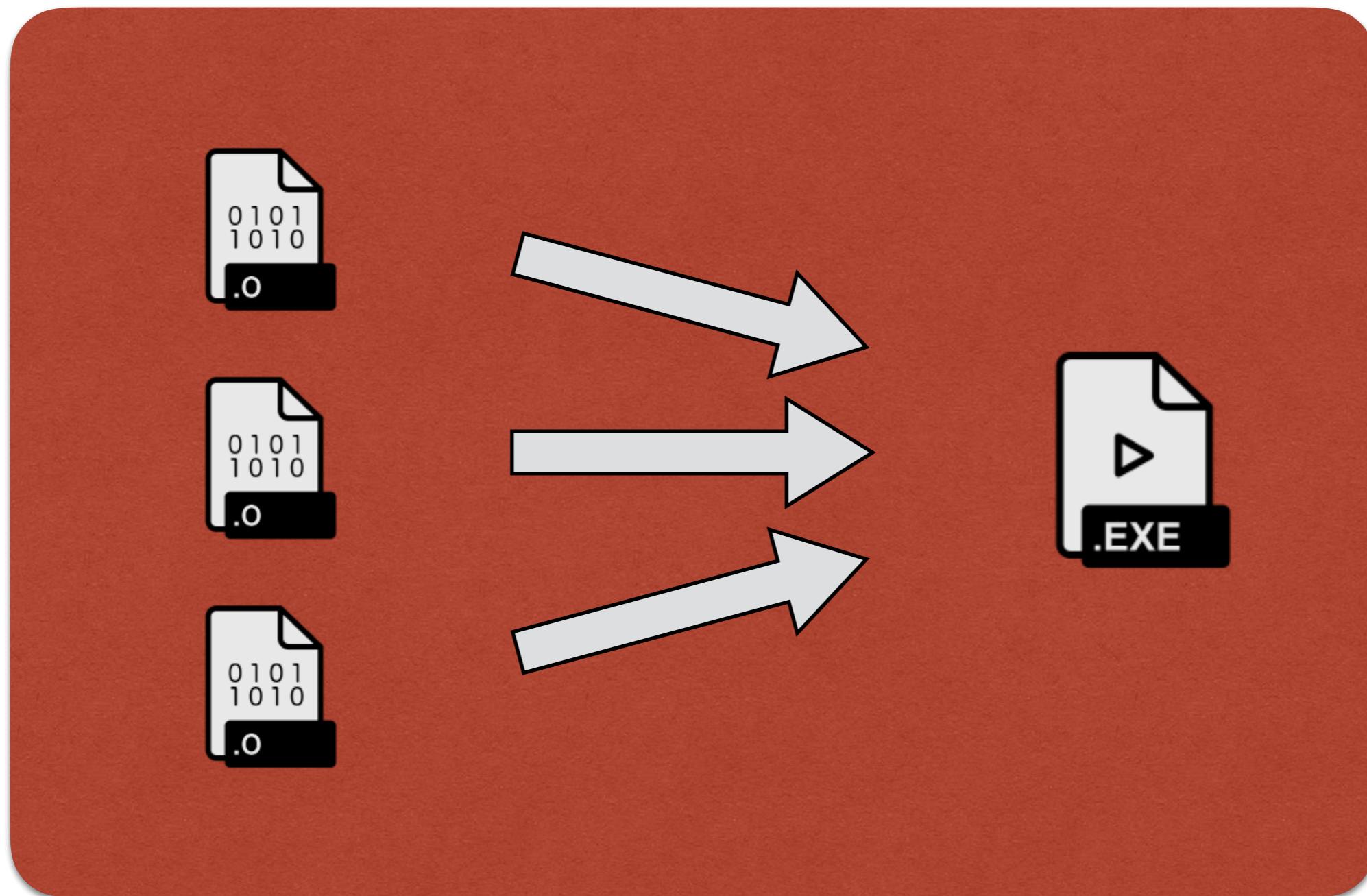


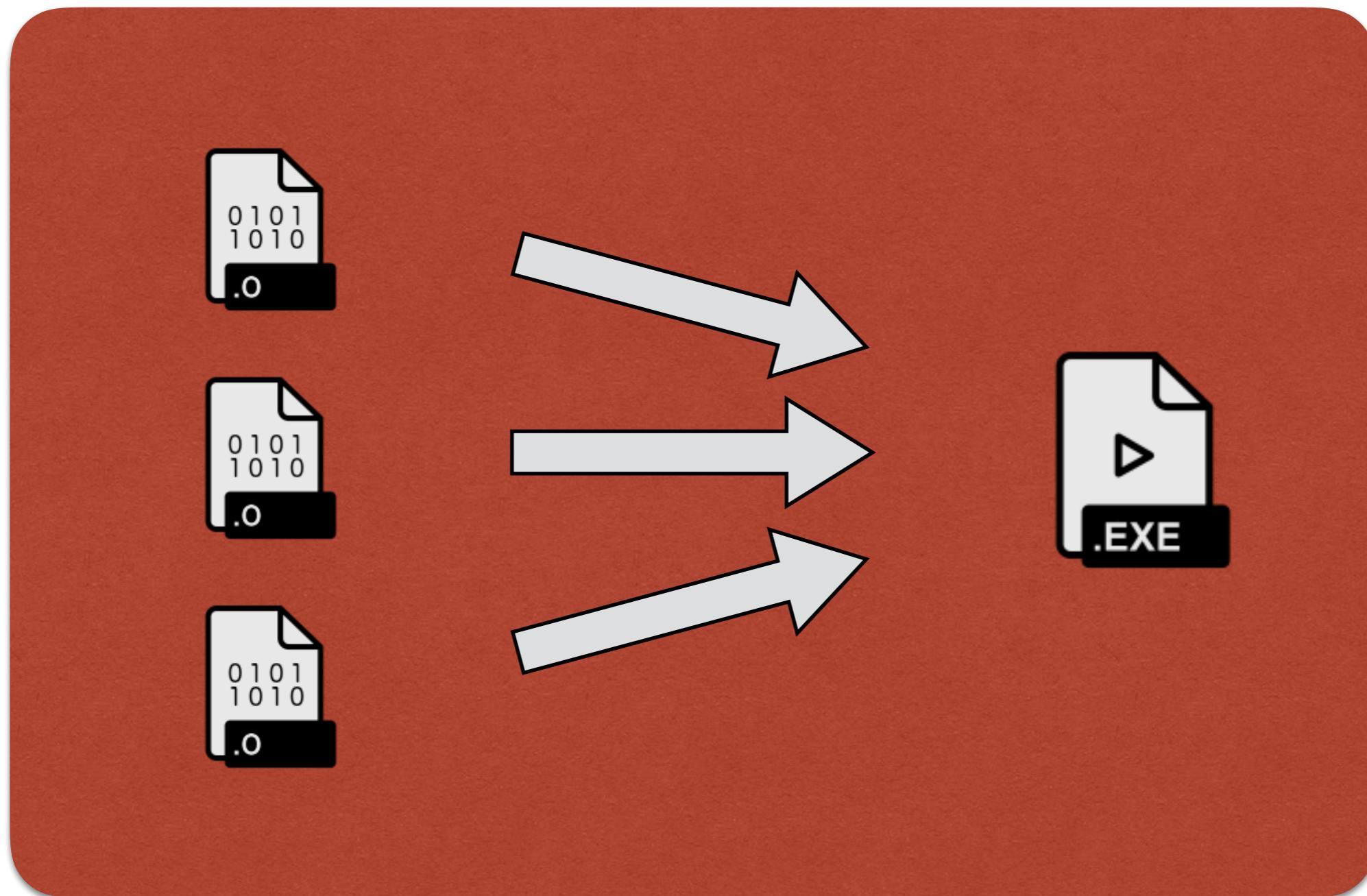
c++filt __Z20lets_get_exponentialP4PairIS_IS_IS_IS_IS_IS_IS_IS_IiiES0_ES1_ES2_ES3_ES4_ES5_ES6_ES7_E



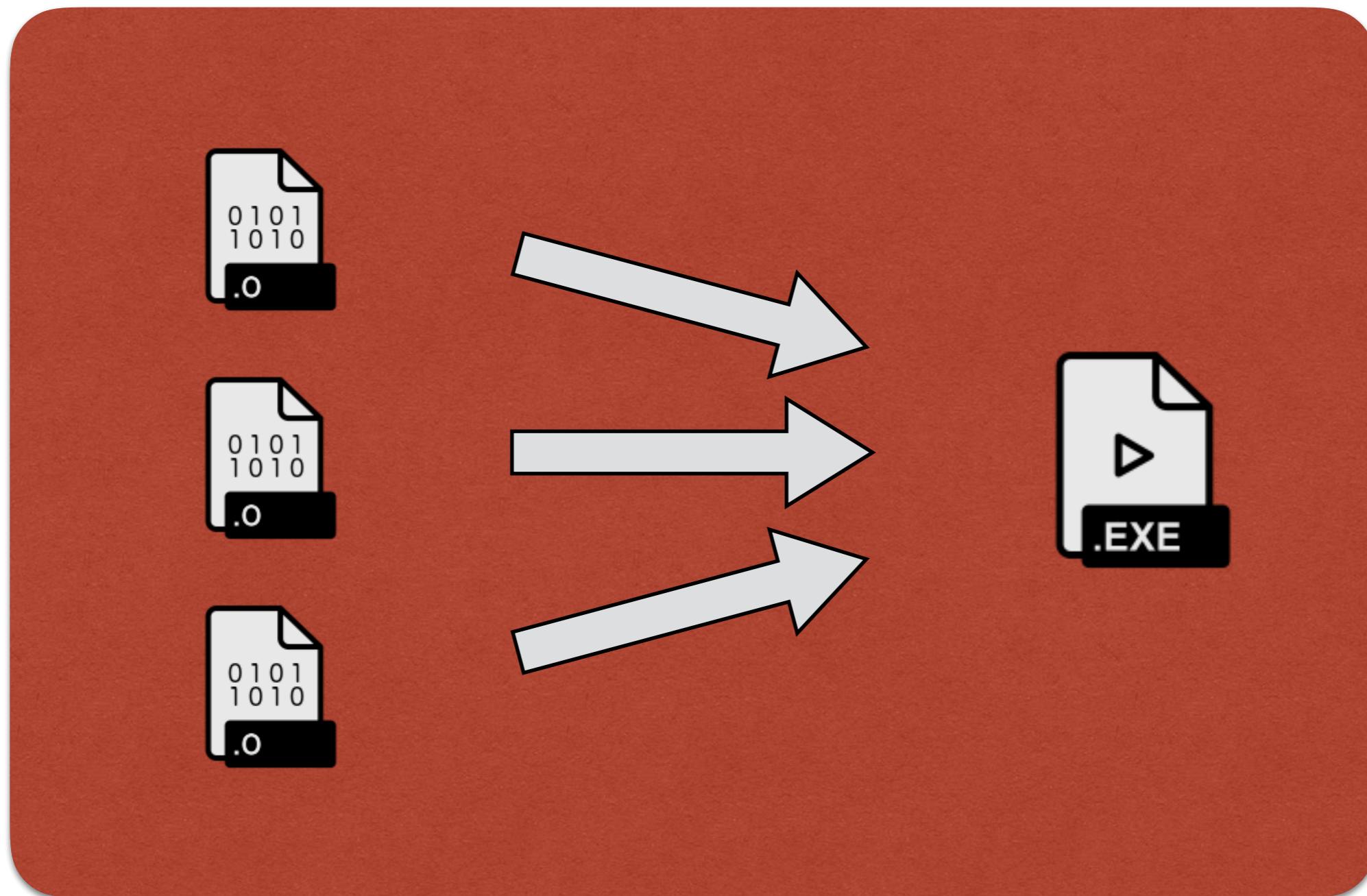
c++filt __Z20lets_get_exponentialIP4PairIS_IS_IS_IS_IS_IS_IS_IS_liiES0_ES1_ES2_ES3_ES4_ES5_ES6_ES7_E

c++filt __Z20lets_get_exponentialP4PairIS_IS_IS_IS_IS_IS_IS_IS_IiiES0_ES1_ES2_ES3_ES4_ES5_ES6_ES7_E

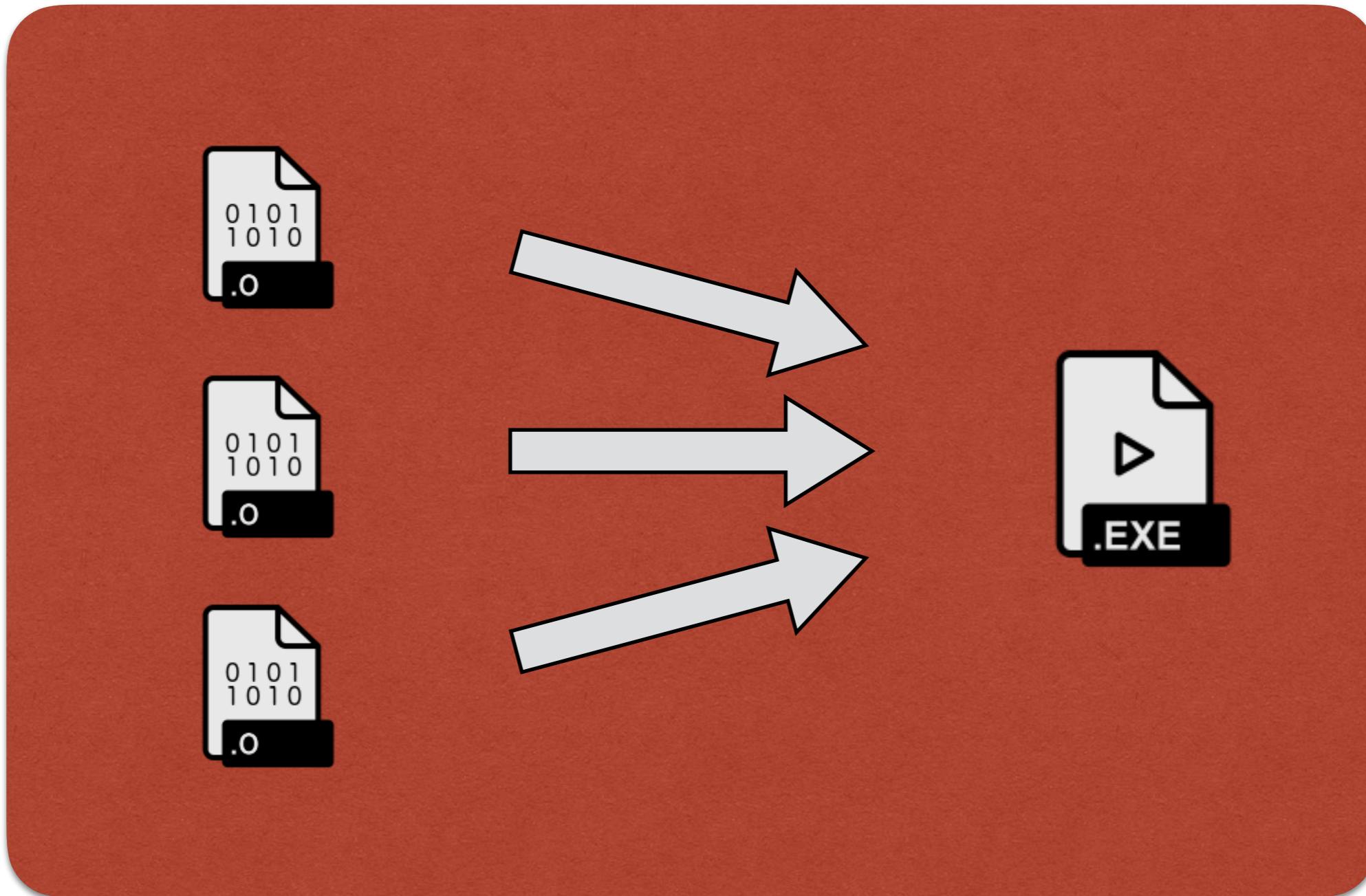


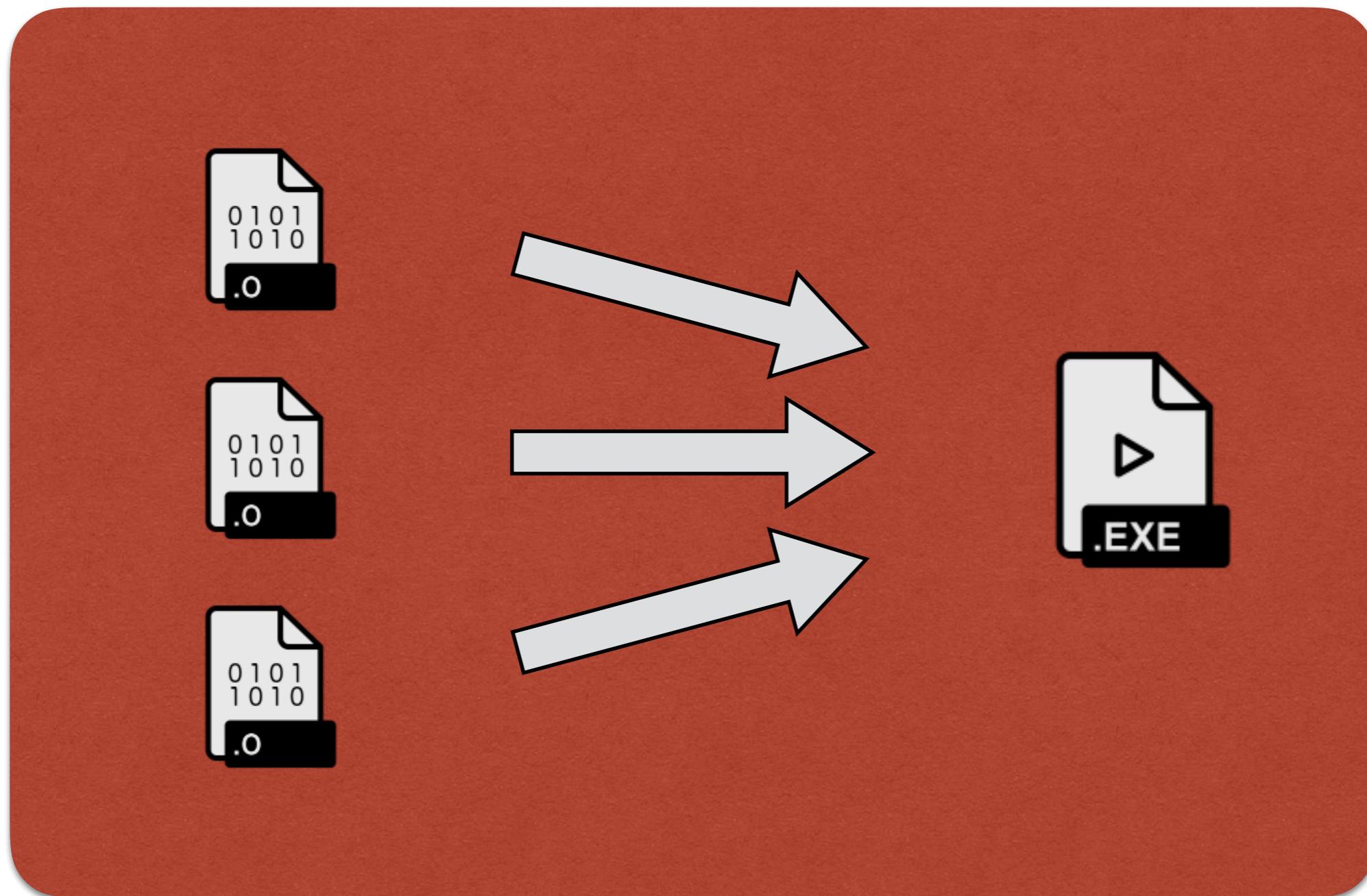


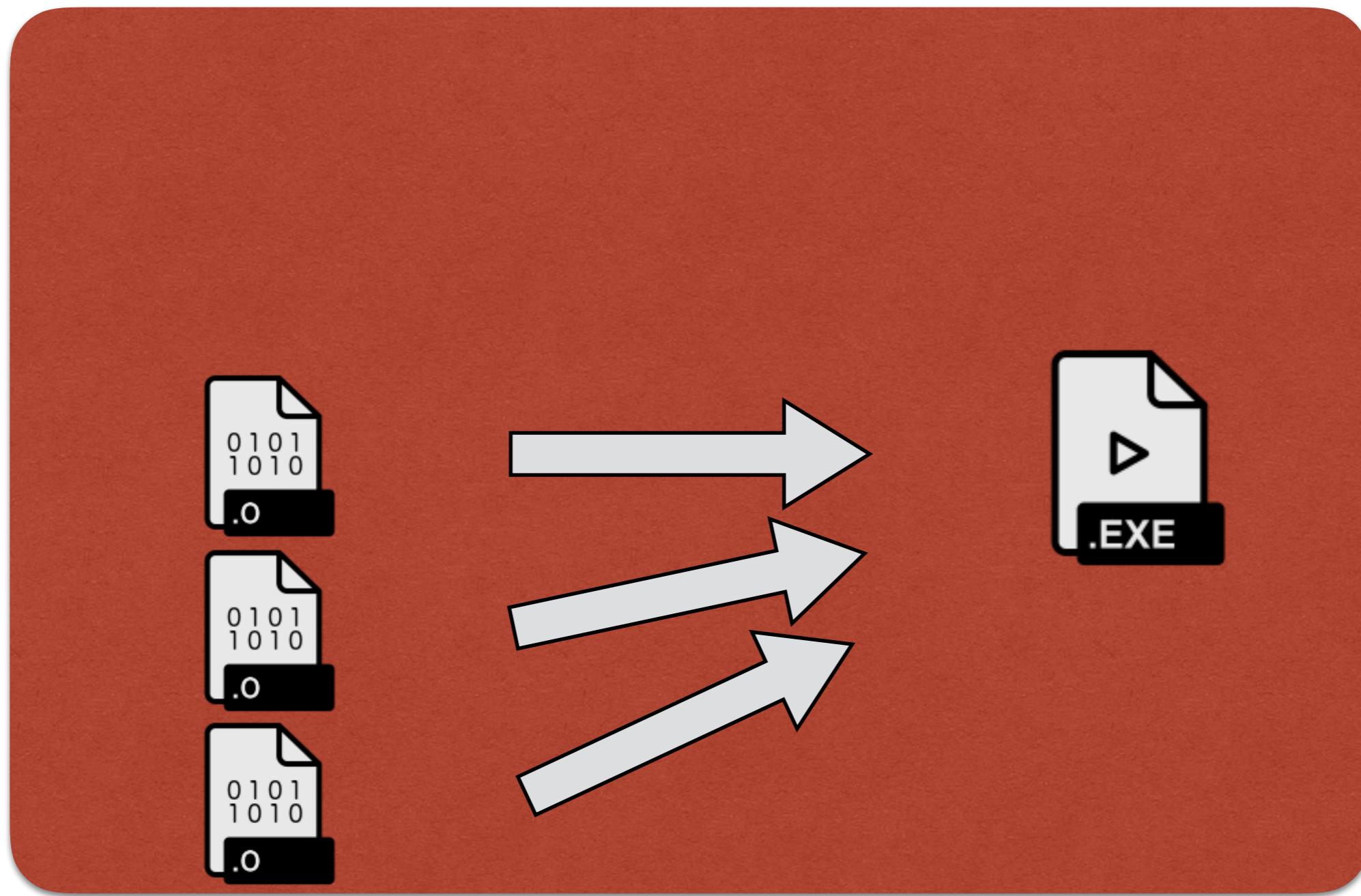


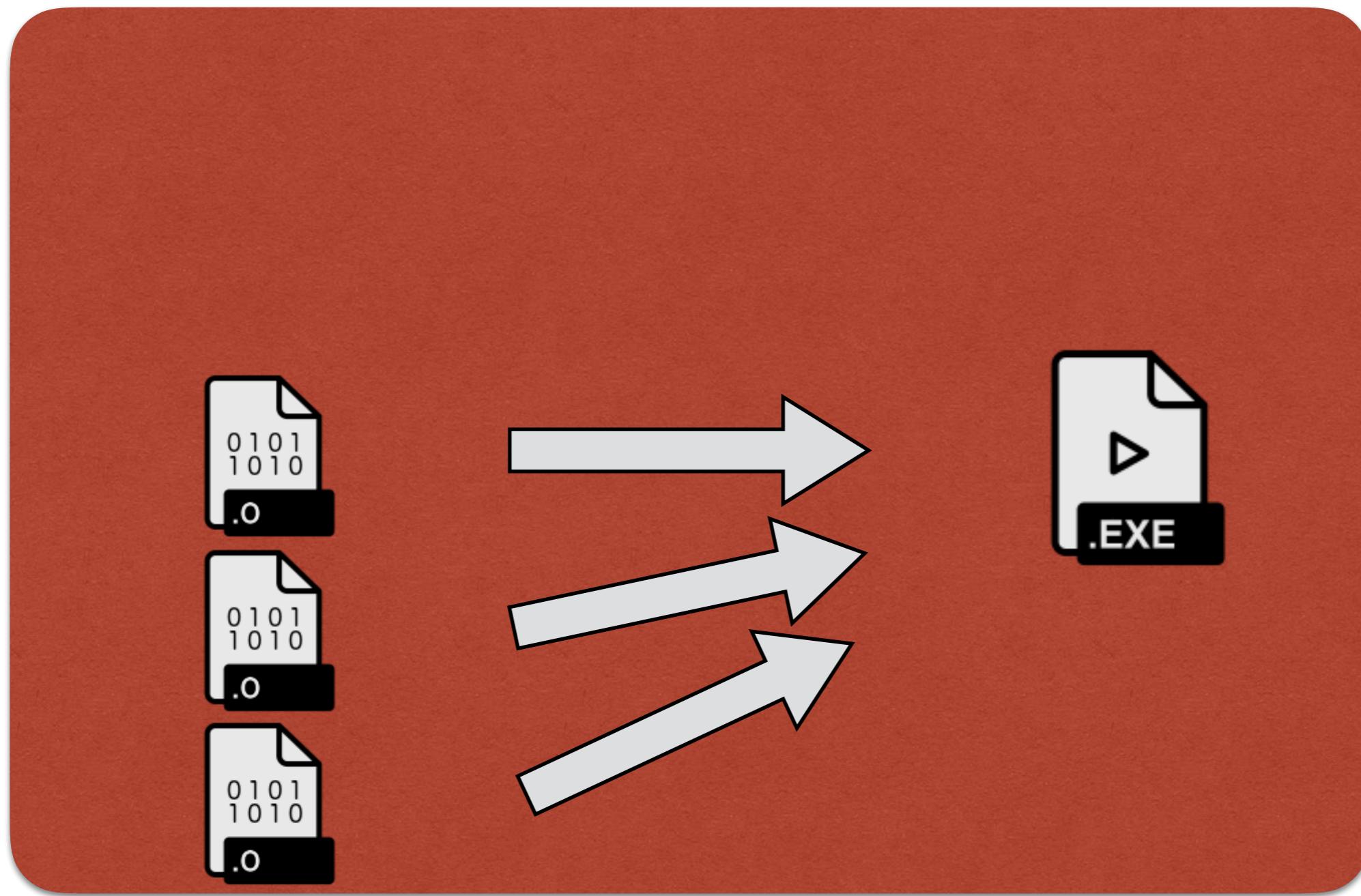


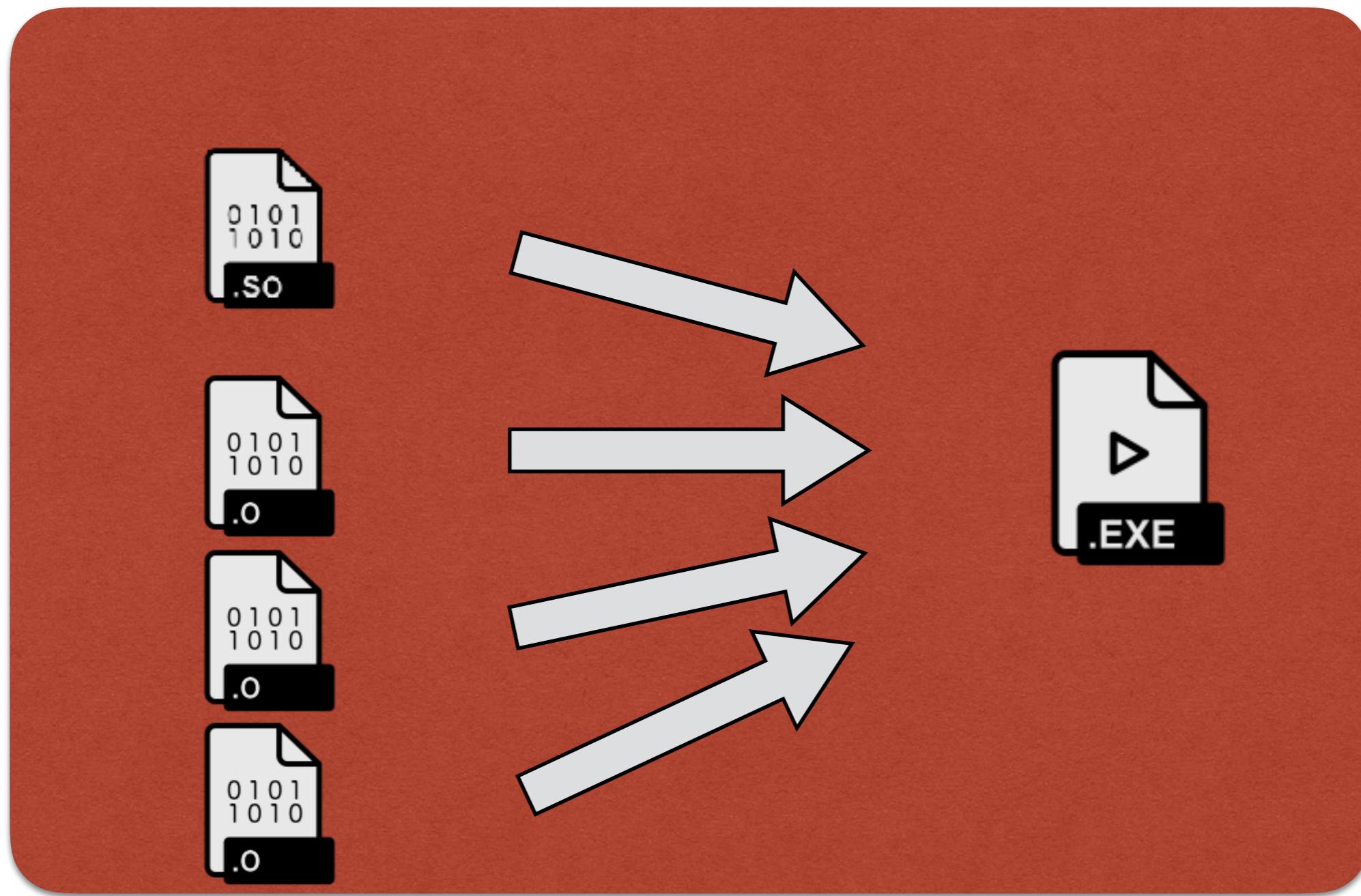
```
clang++ main.o CloudBasedSuperAI.o -o main
```



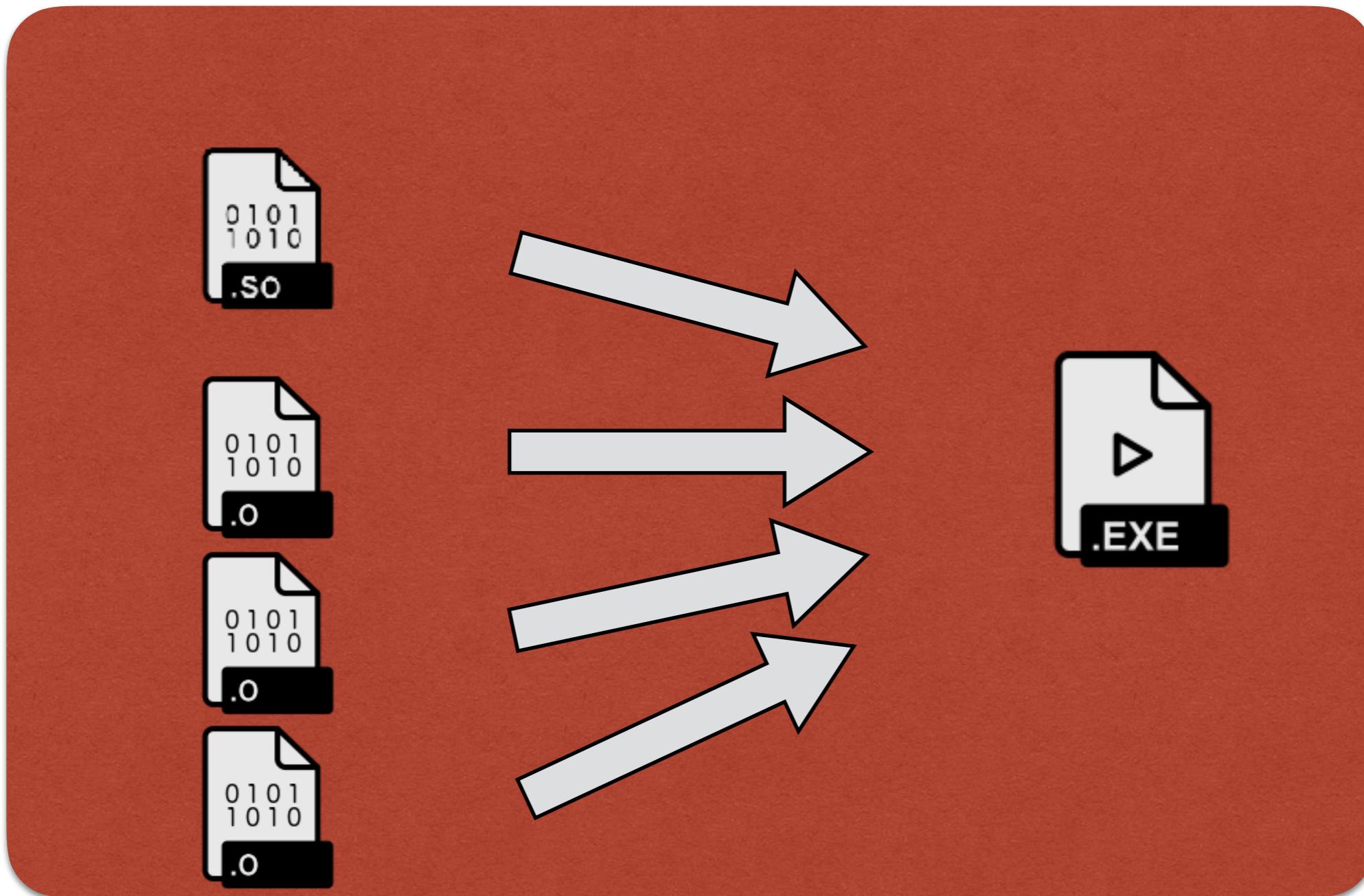




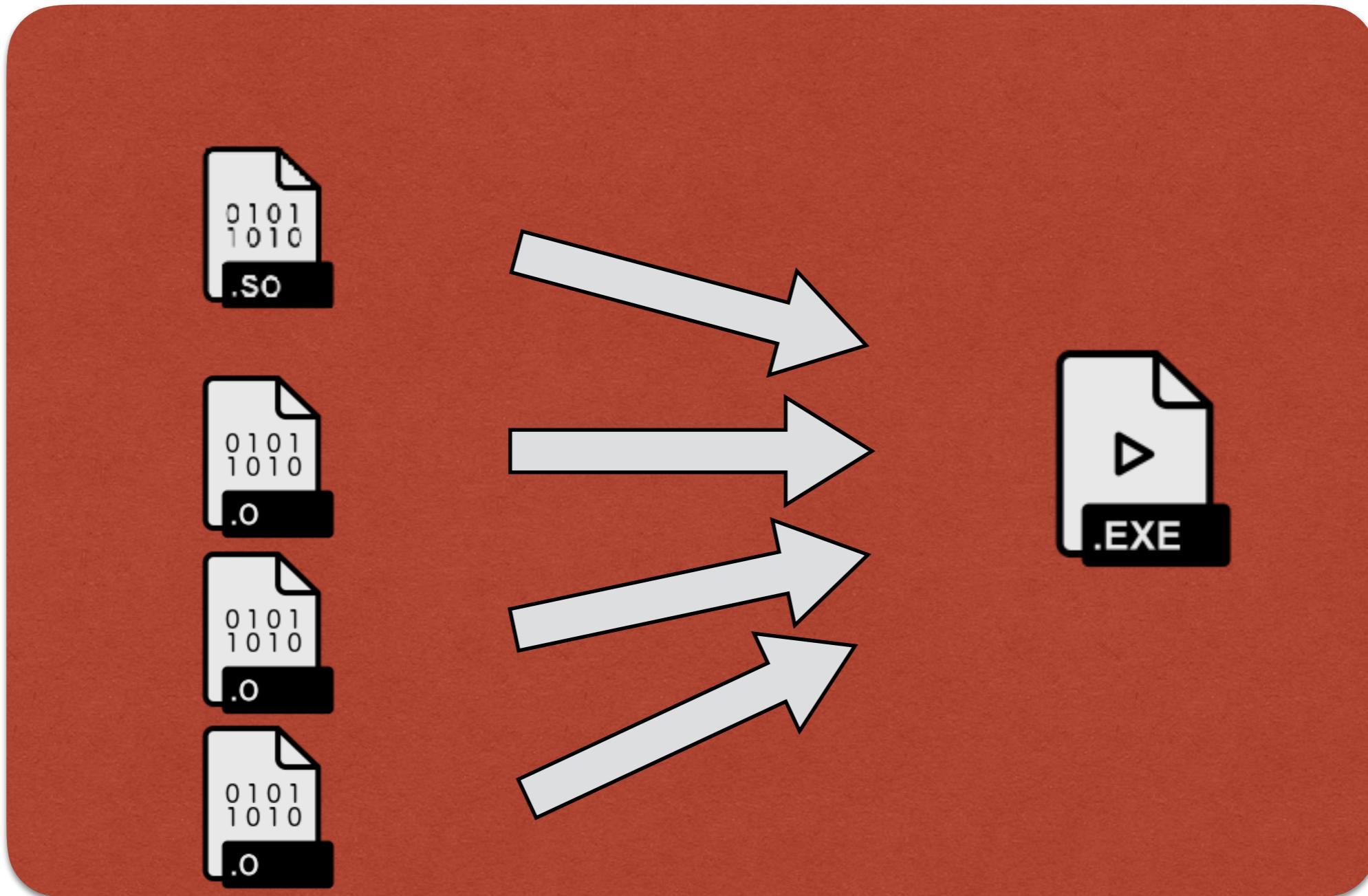




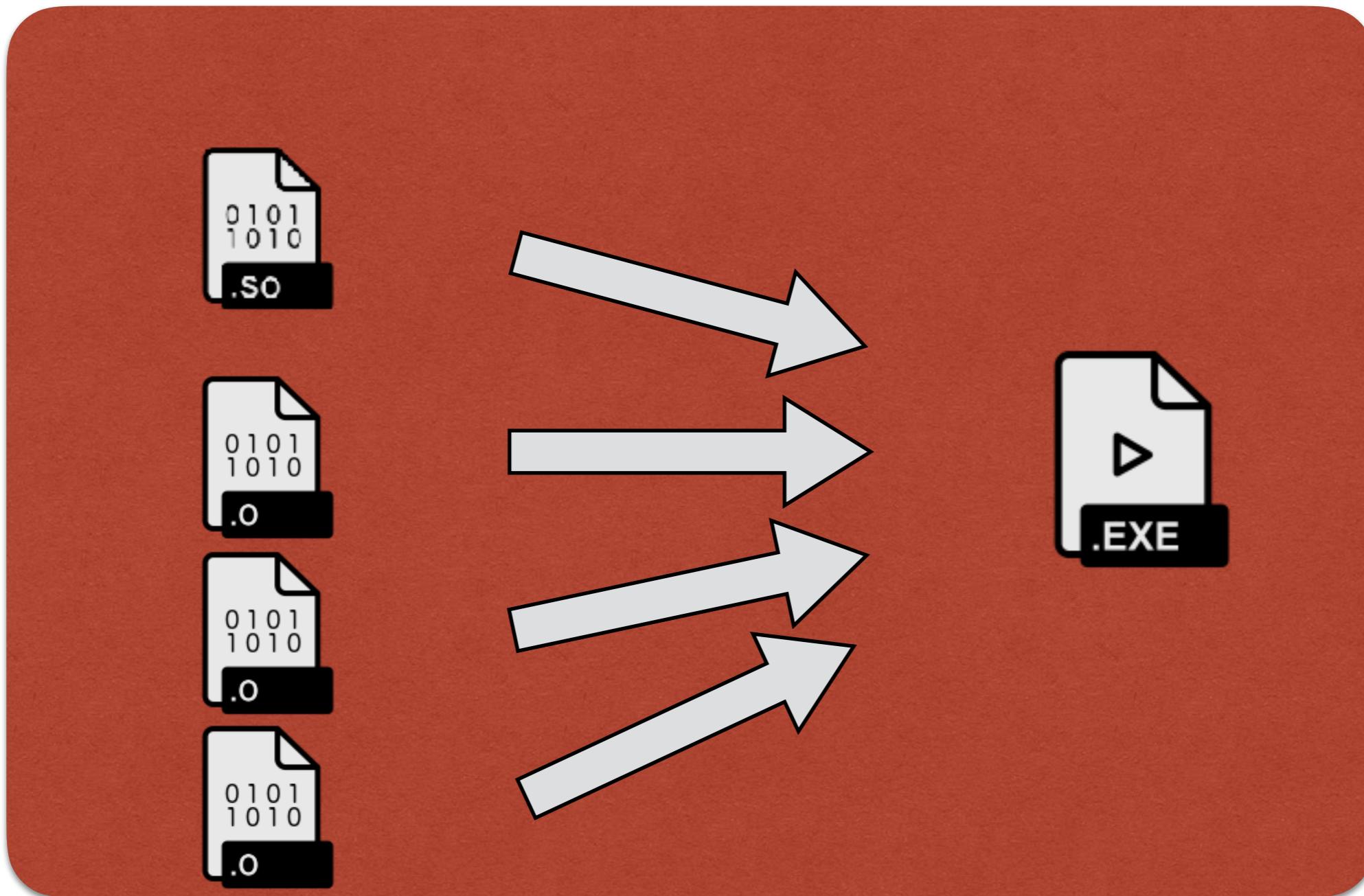
```
clang++ main.o CloudBasedSuperAI.o -Xlinker -t -o main
```



clang++ main.o CloudBasedSuperAI.o -Xlinker -t -o main



```
clang++ main.o CloudBasedSuperAI.o -Xlinker -t -o main
```



```
clang++ main.o CloudBasedSuperAI.o -Xlinker -t -o main
```

CloudBasedSuperAI.o

main.o

`${SDKPath}/usr/lib/libSystem.tbd`

`${SDKPath}/usr/lib/libc++.tbd`

`${SDKPath}/usr/lib/system/libcache.tbd`

`${SDKPath}/usr/lib/system/libcommonCrypto.tbd`

`${SDKPath}/usr/lib/system/libcompiler_rt.tbd`

`${SDKPath}/usr/lib/system/libcopyfile.tbd`

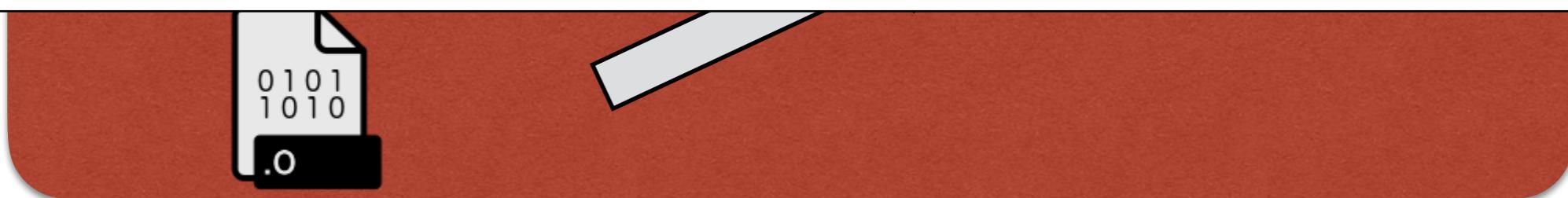
`${SDKPath}/usr/lib/system/libcorecrypto.tbd`

`${SDKPath}/usr/lib/system/libdispatch.tbd`

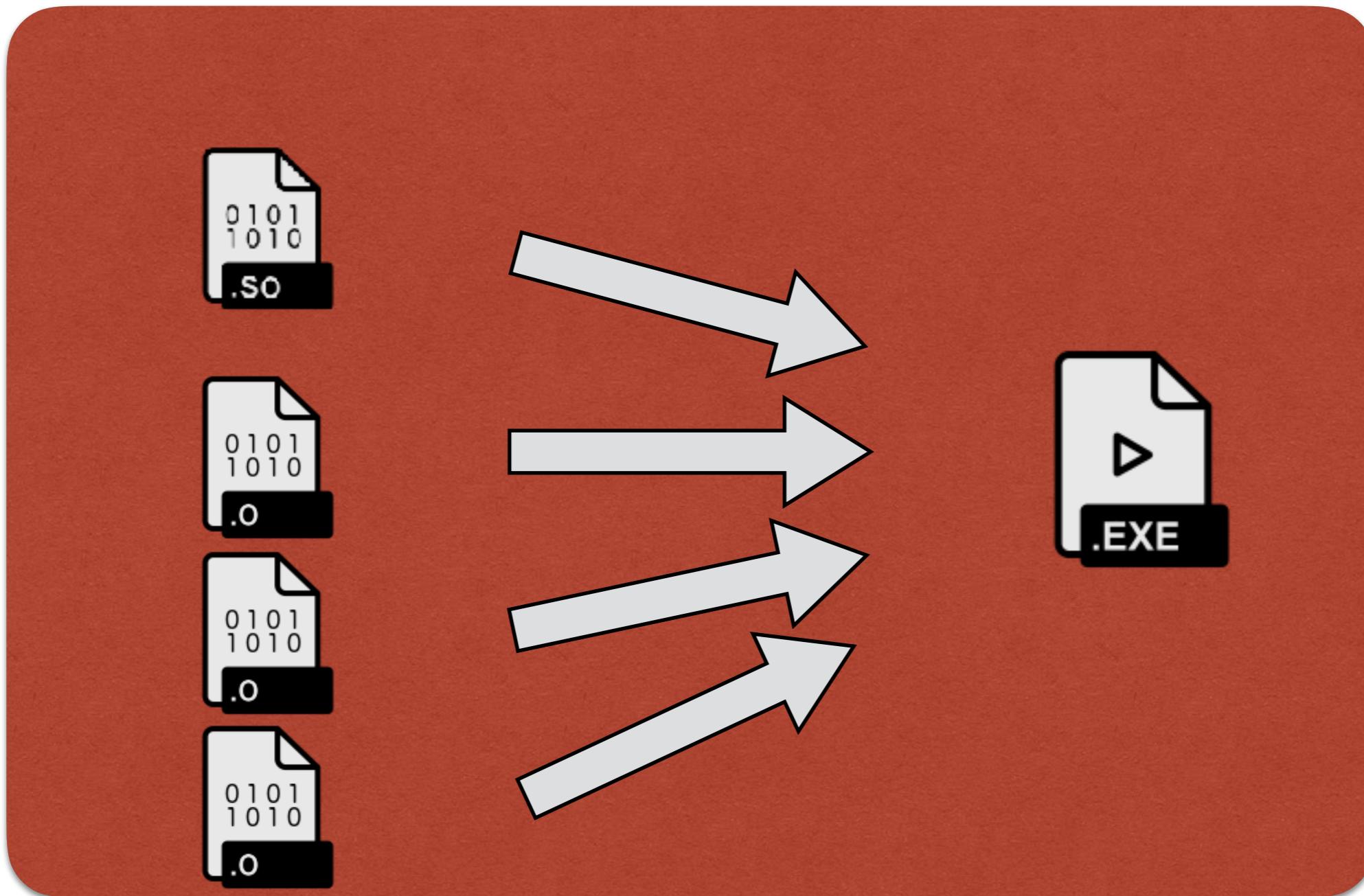
`${SDKPath}/usr/lib/system/libdyld.tbd`

`${SDKPath}/usr/lib/system/libunwind.tbd`

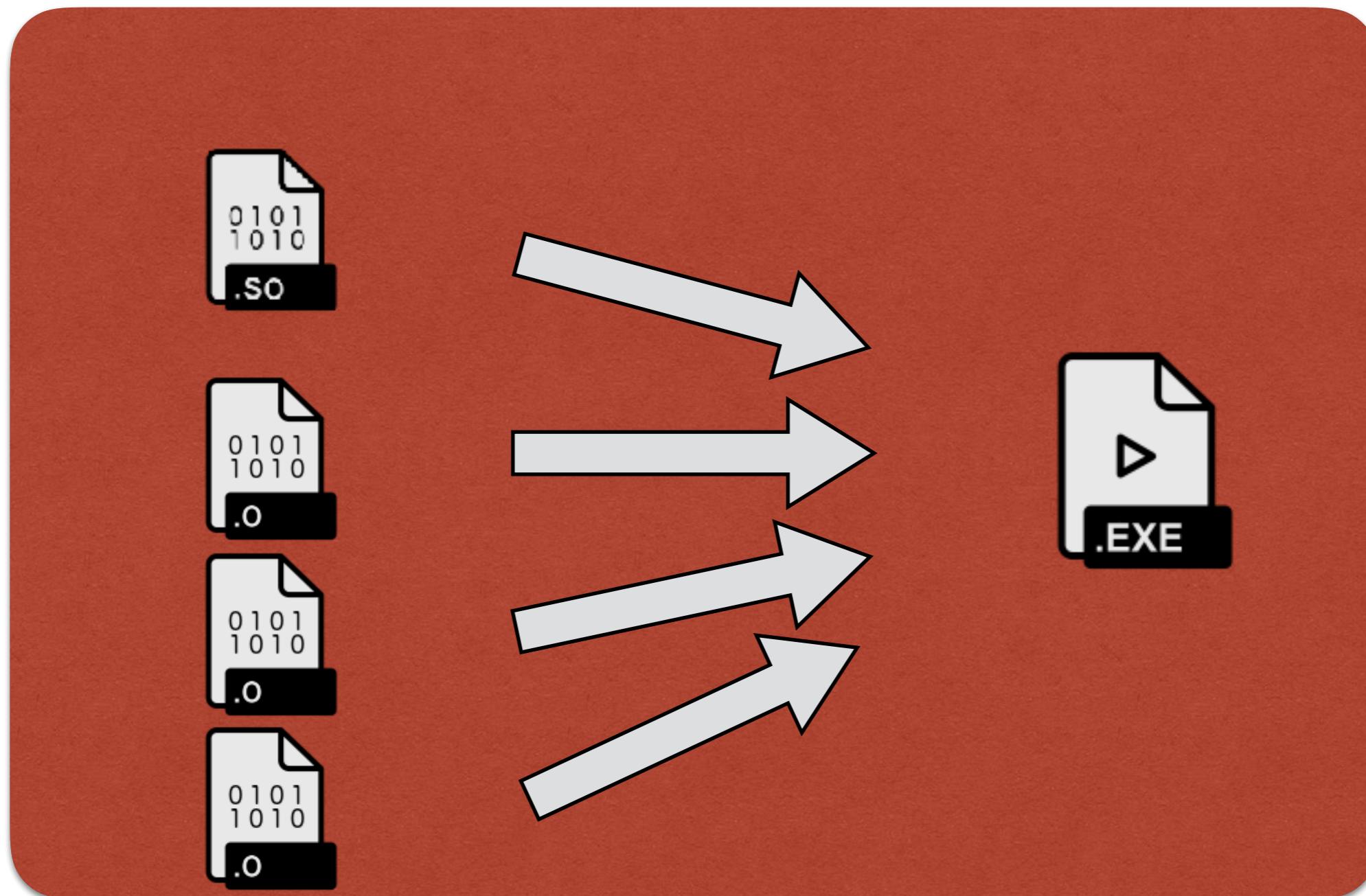
`${SDKPath}/usr/lib/system/libxpc.tbd`



```
clang++ main.o CloudBasedSuperAI.o -Xlinker -t -o main
```

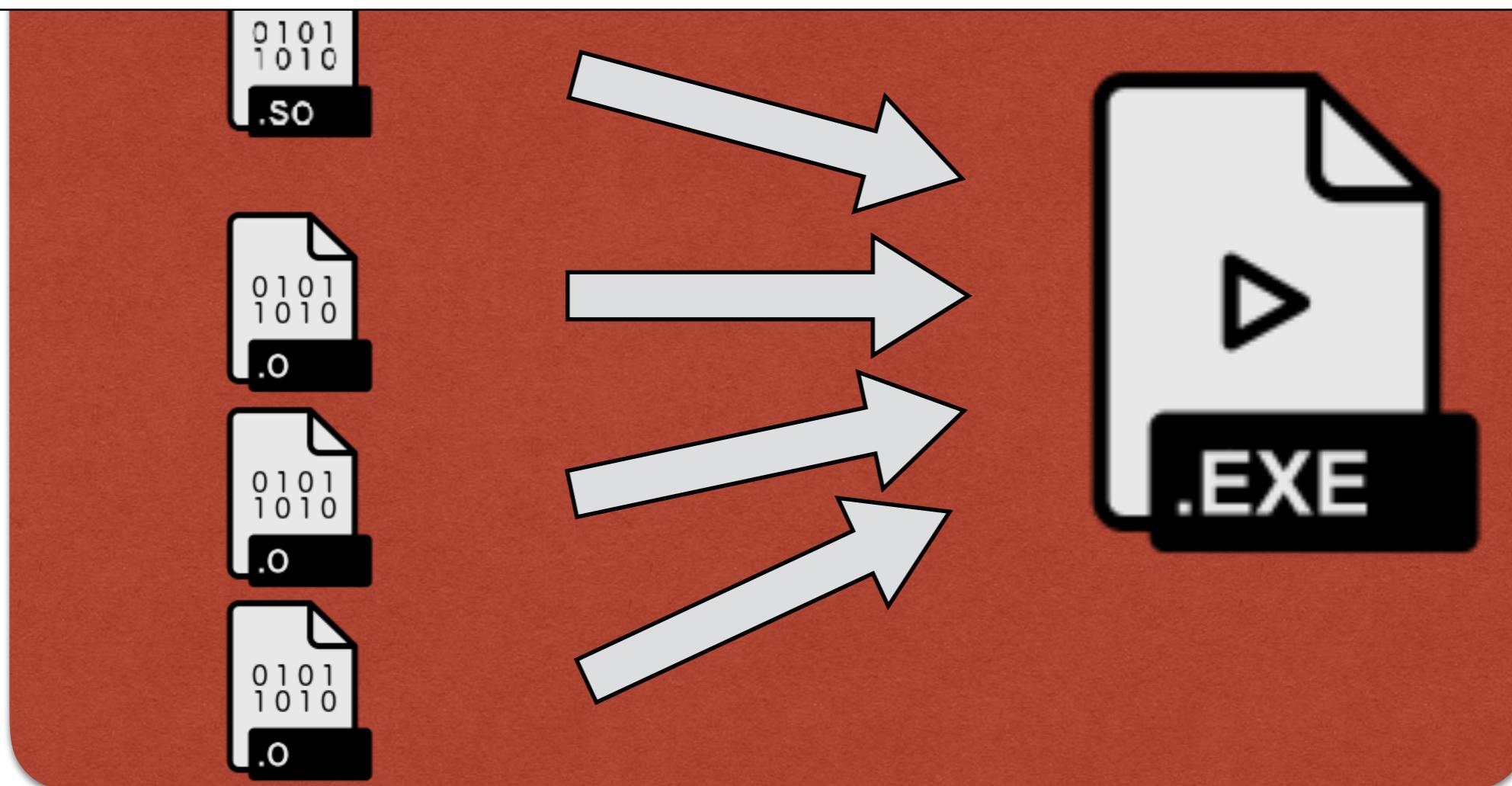


nm main

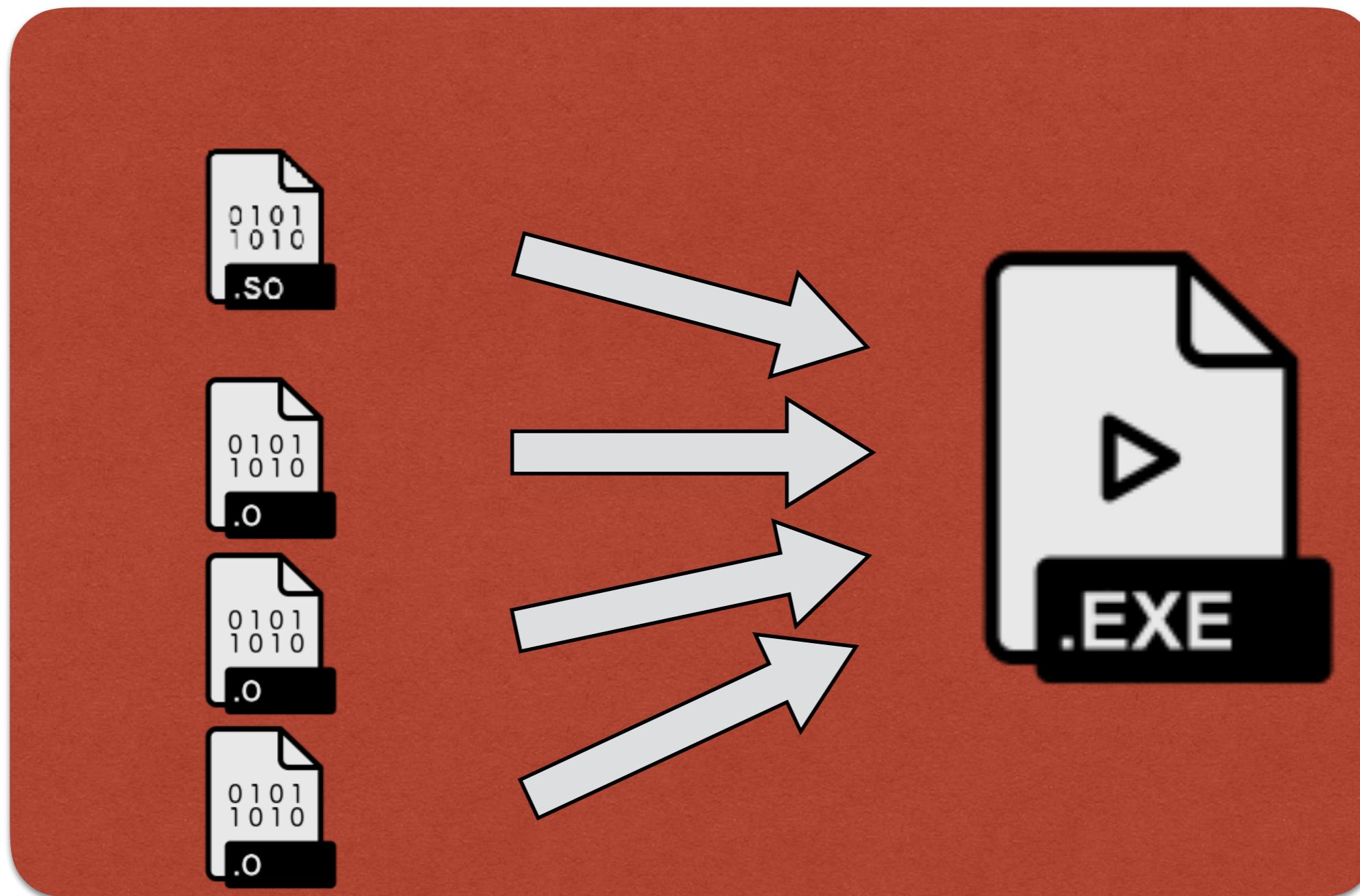


nm main

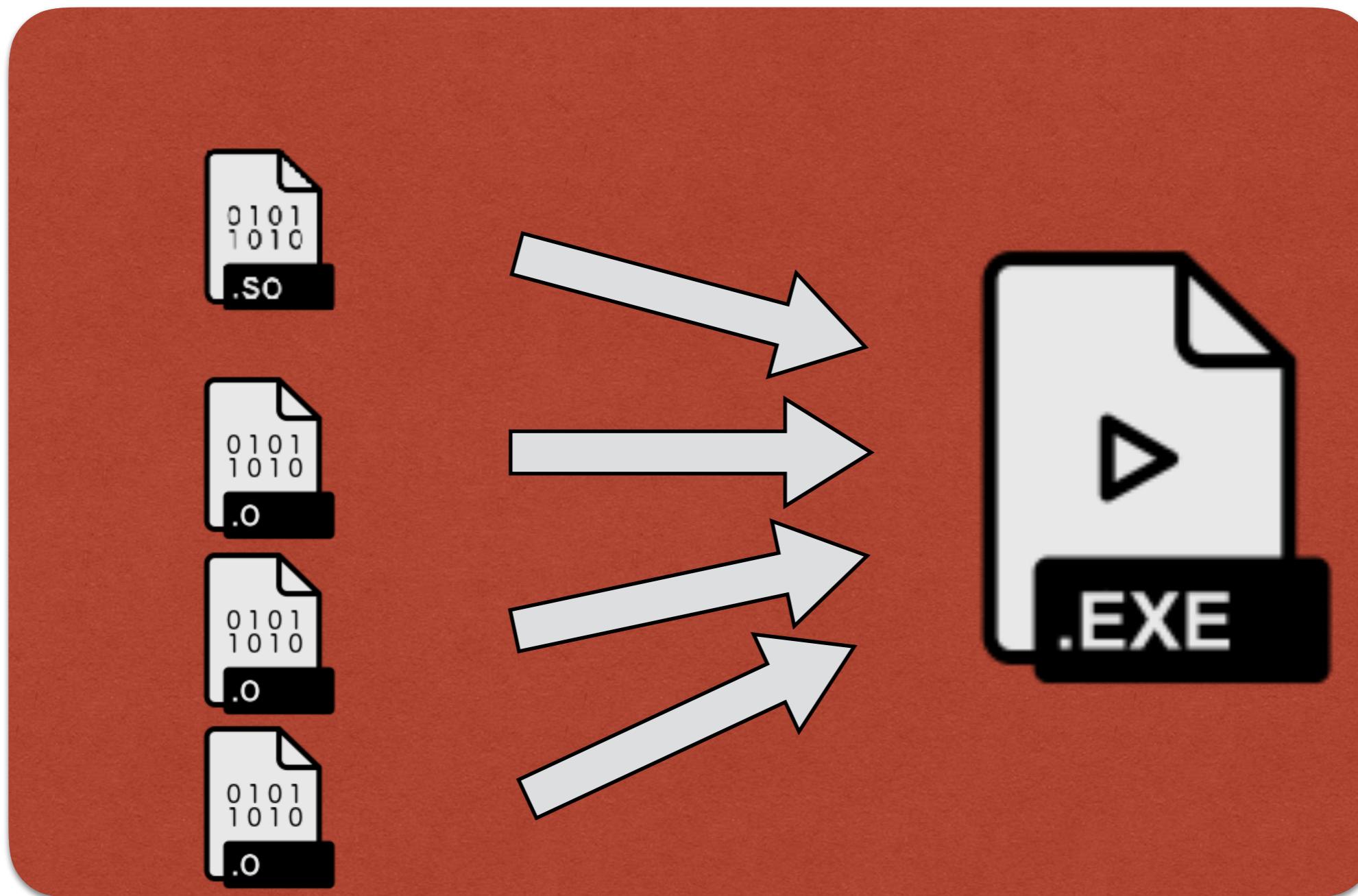
```
0000000100000fa0 T __Z20runCloudBasedSuperAIv
0000000100000fb0 t __Z23aiPrivateImplementationv
0000000100000000 T __mh_execute_header
0000000100000f80 T _main
U dyld_stub_binder
```



nm main

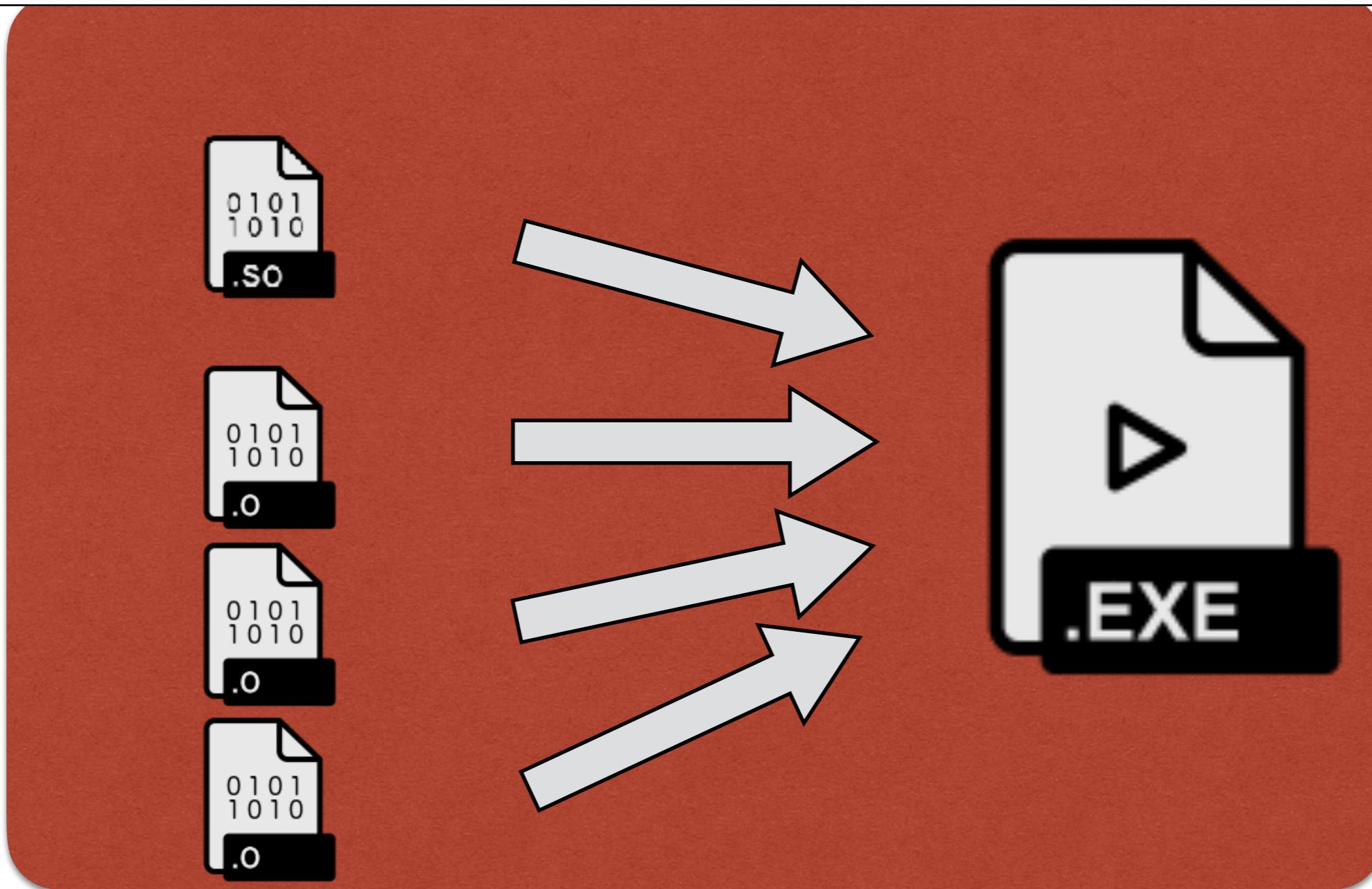


strip main; nm main



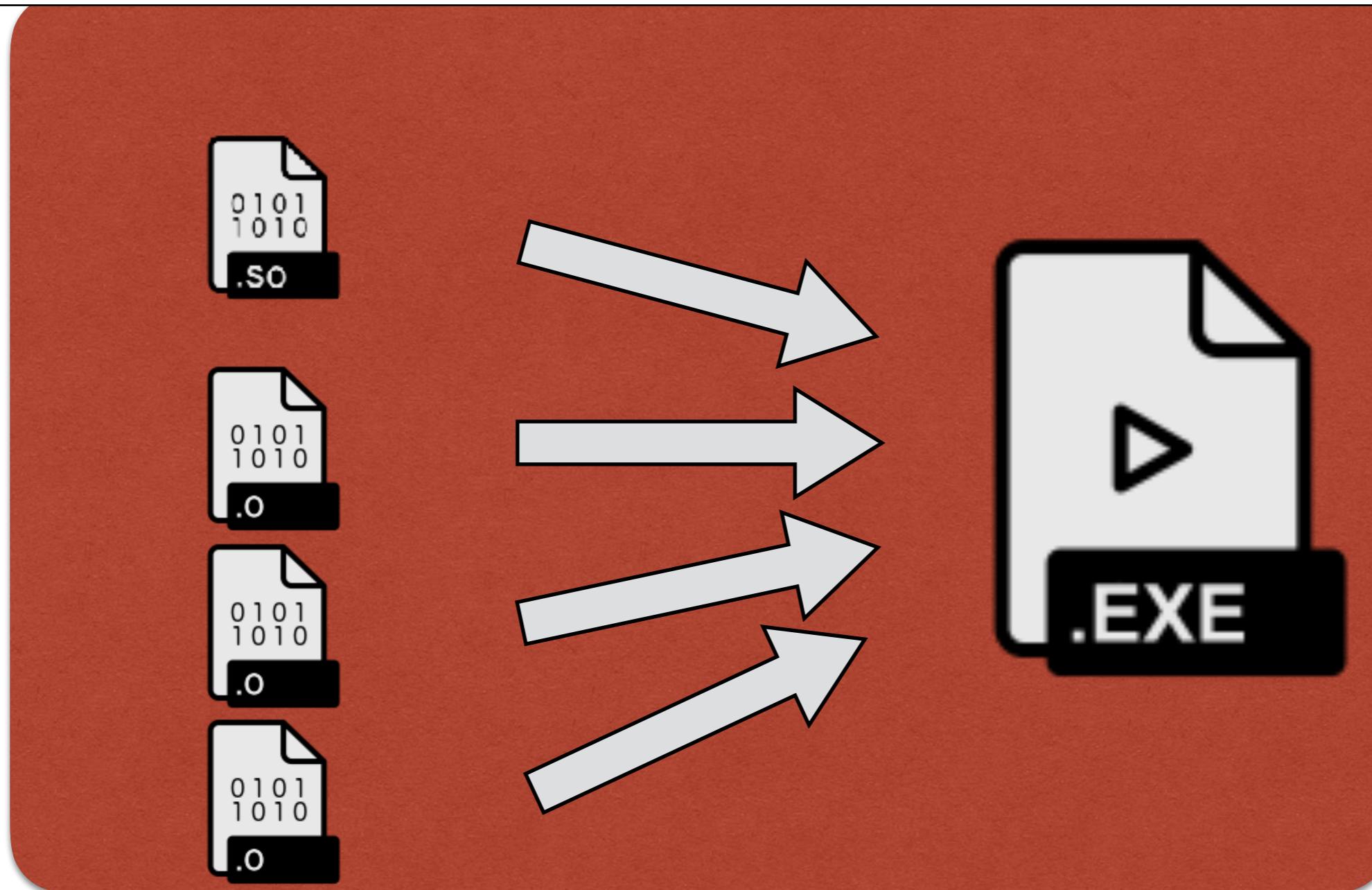
strip main; nm main

0000000100000000 T __mh_execute_header
U dyld_stub_binder



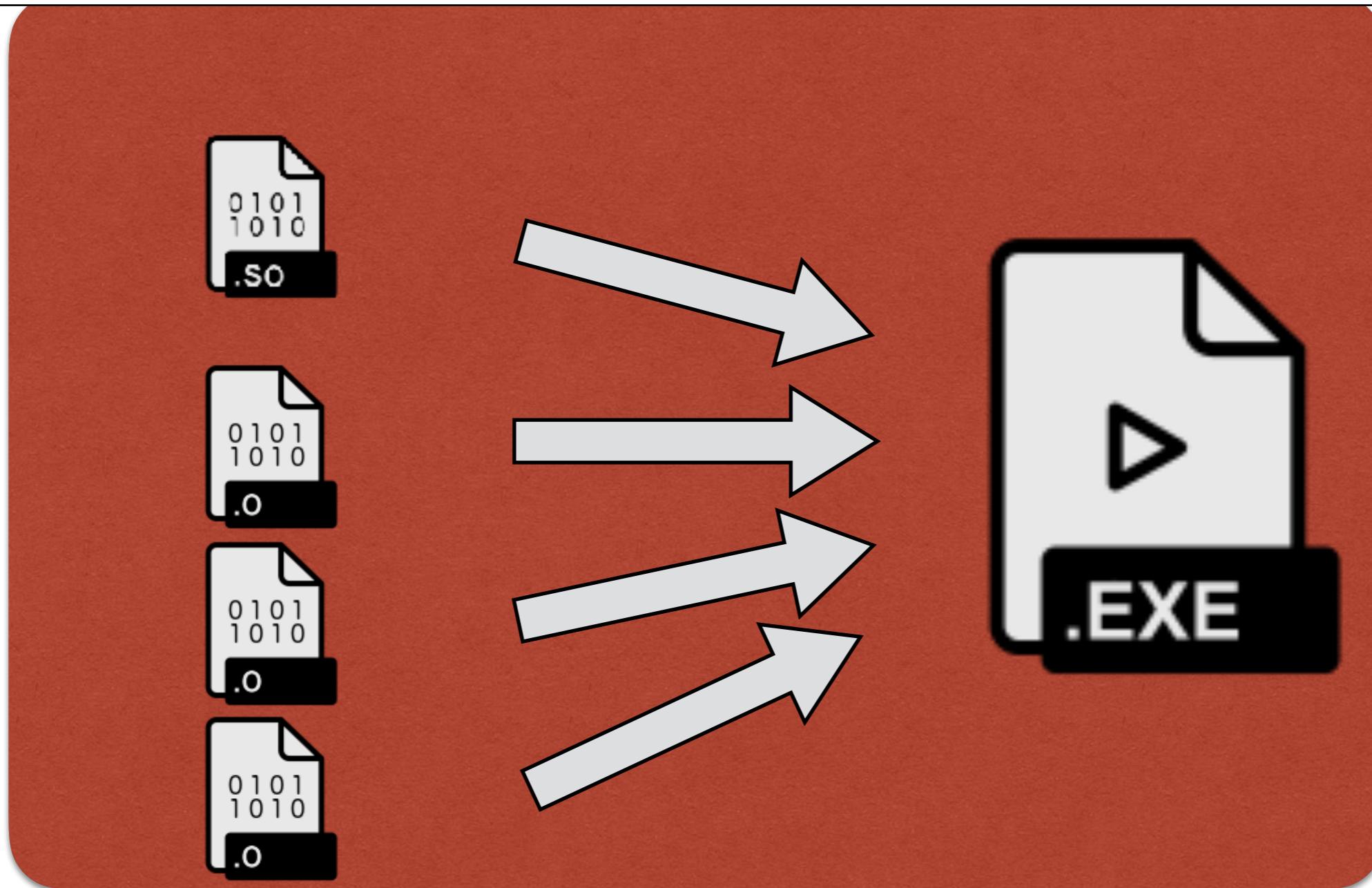
strip main; nm main

0000000100000000 T __mh_execute_header
U dyld_stub_binder

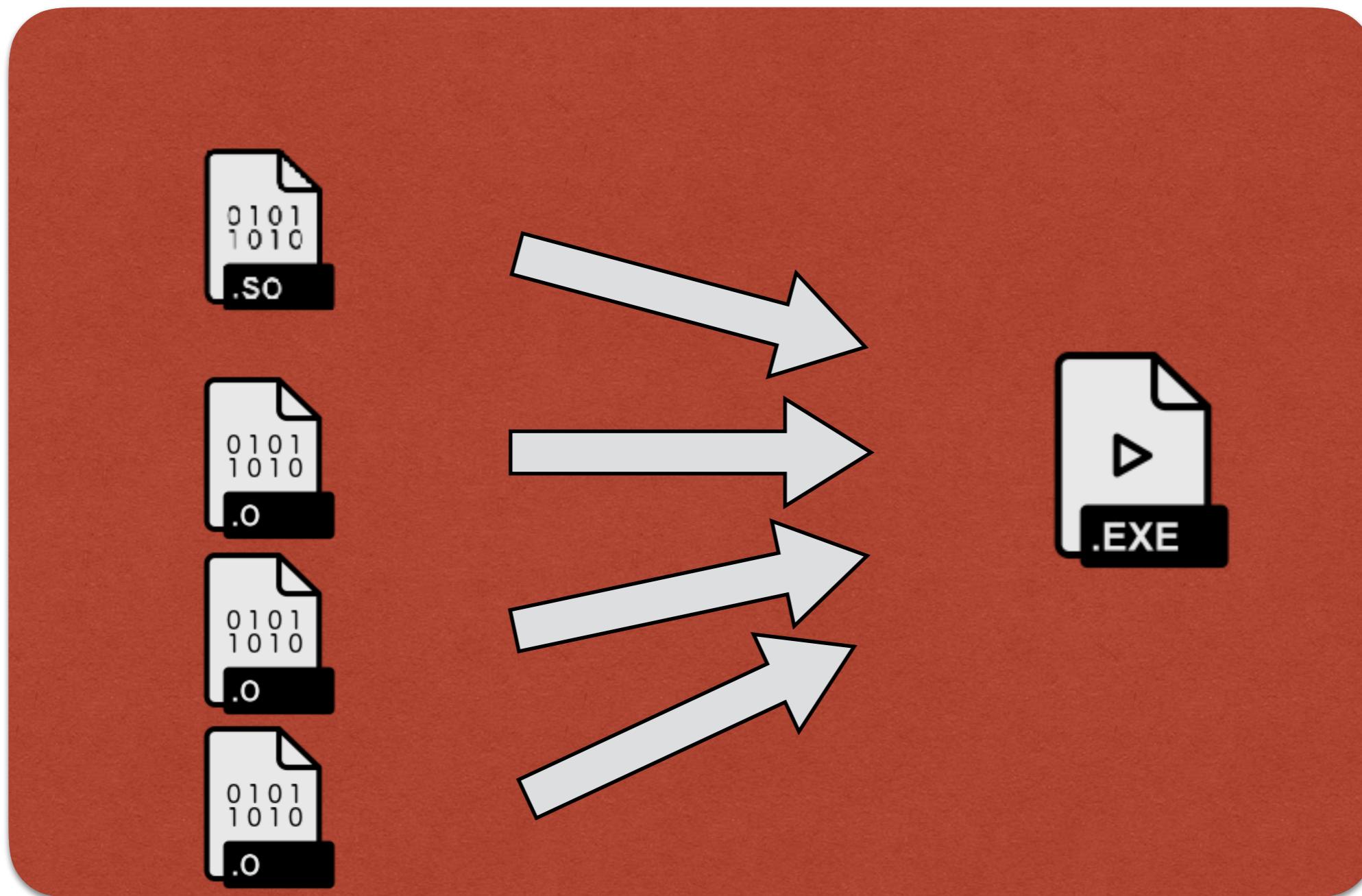


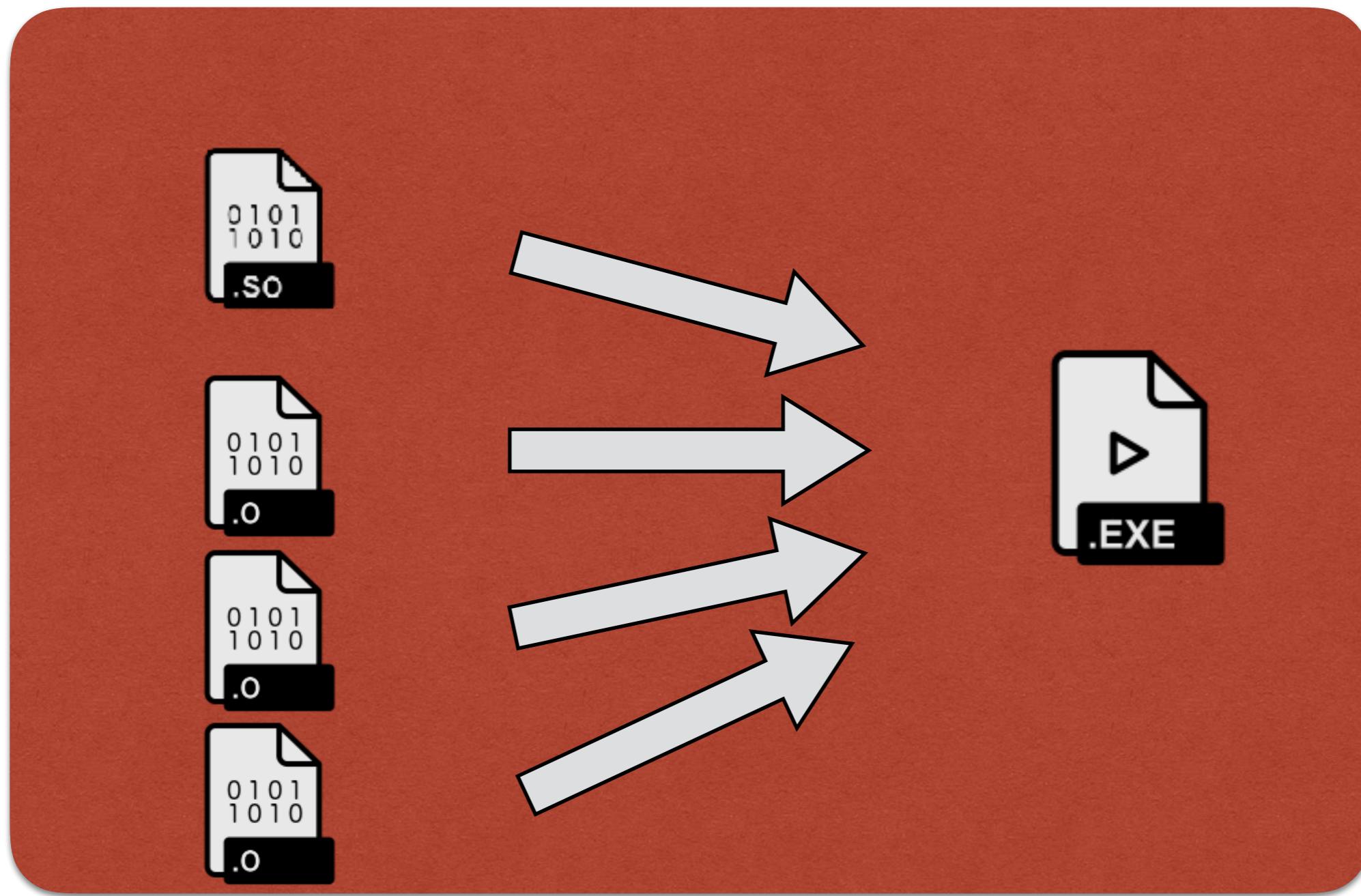
strip main; nm main

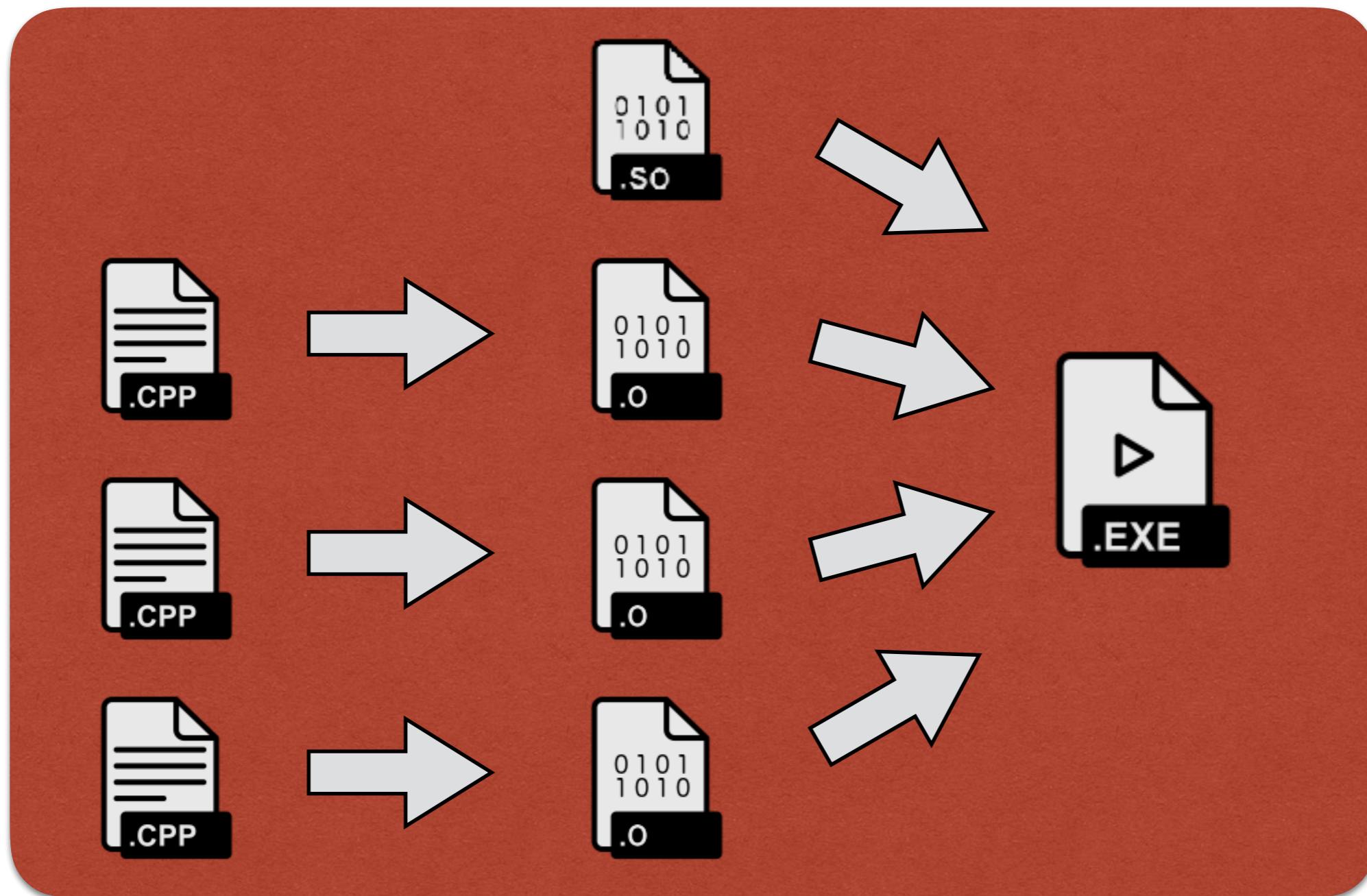
0000000100000000 T __mh_execute_header
U dyld_stub_binder

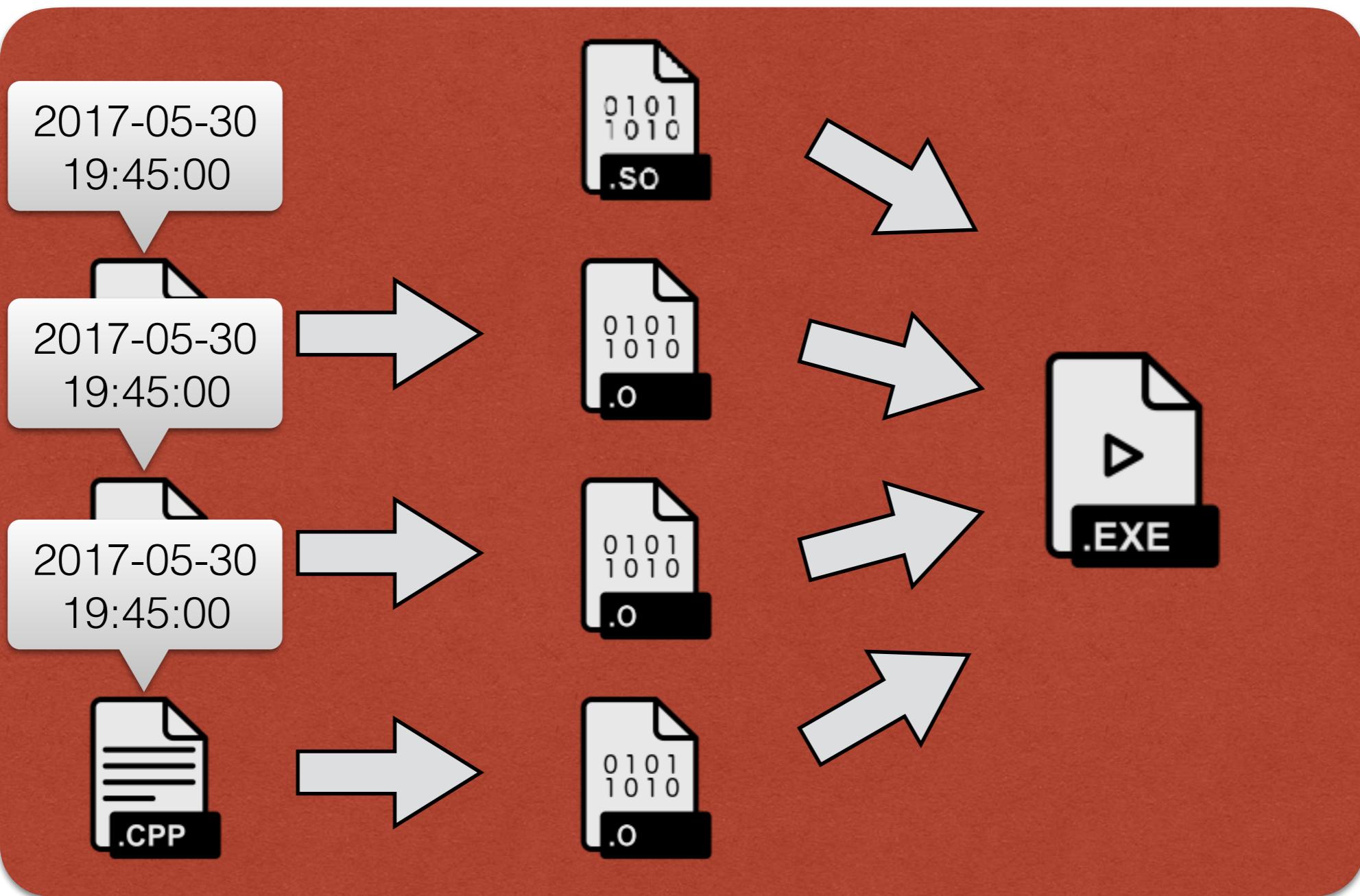


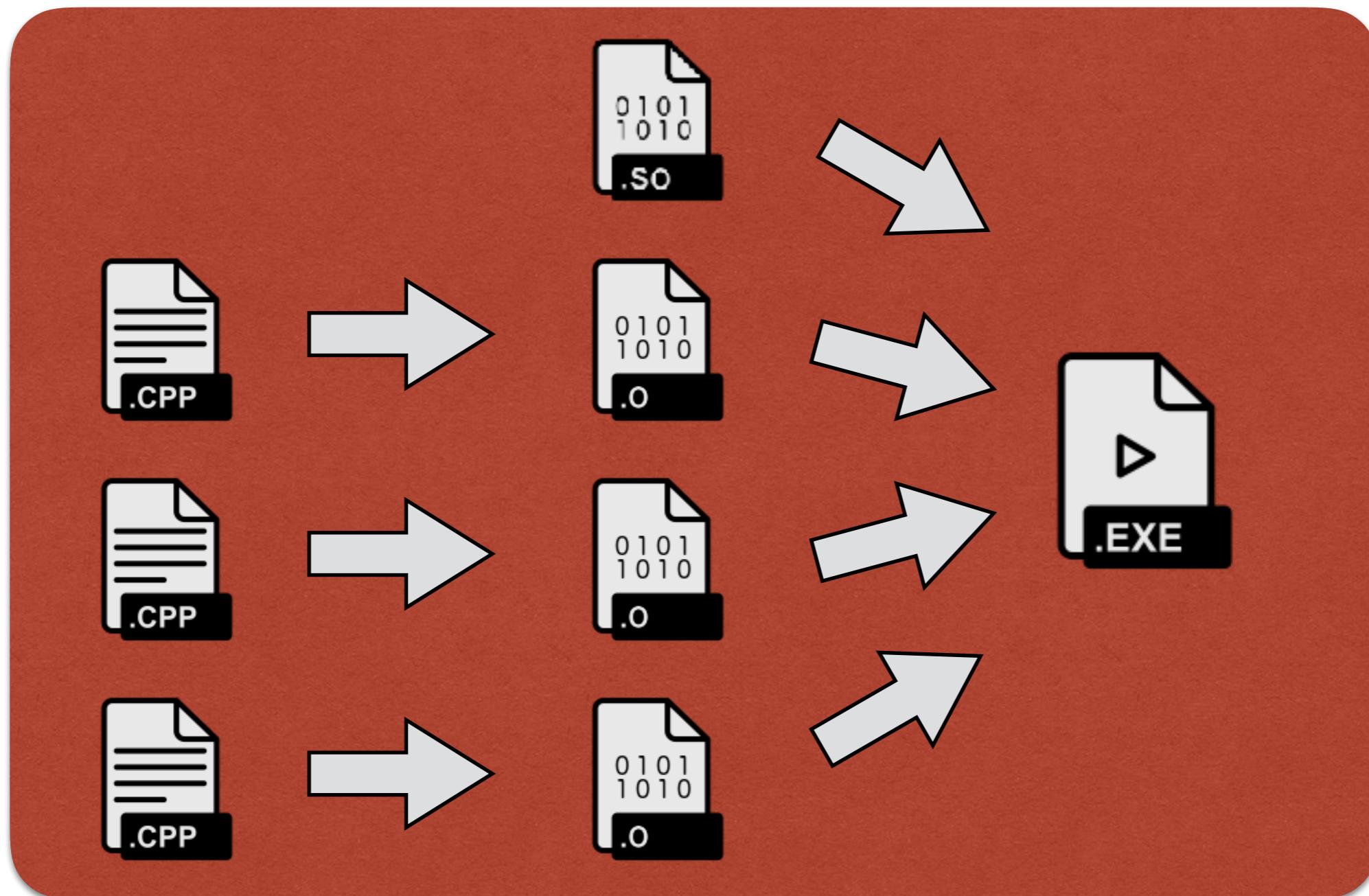
strip main; nm main

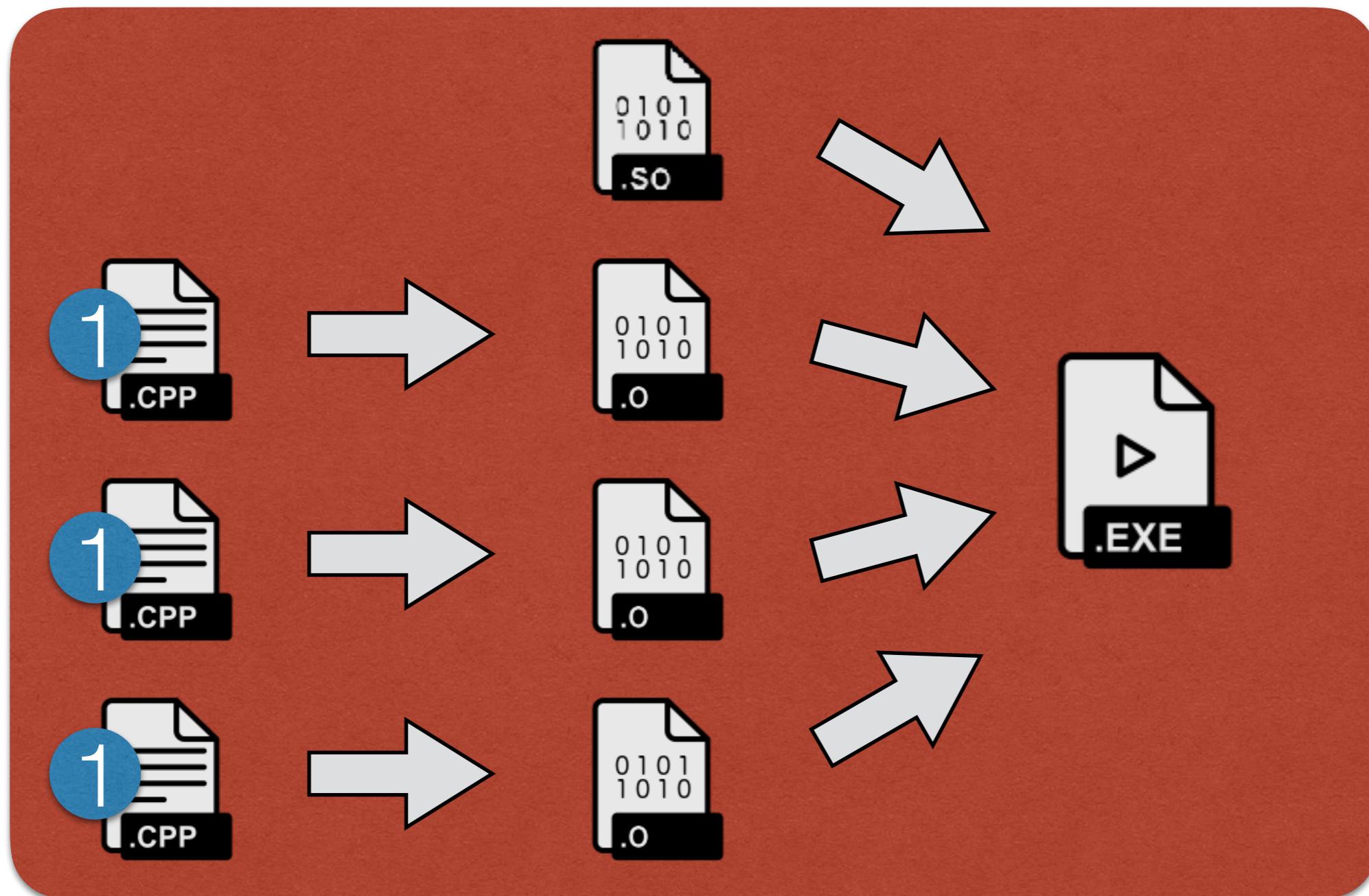


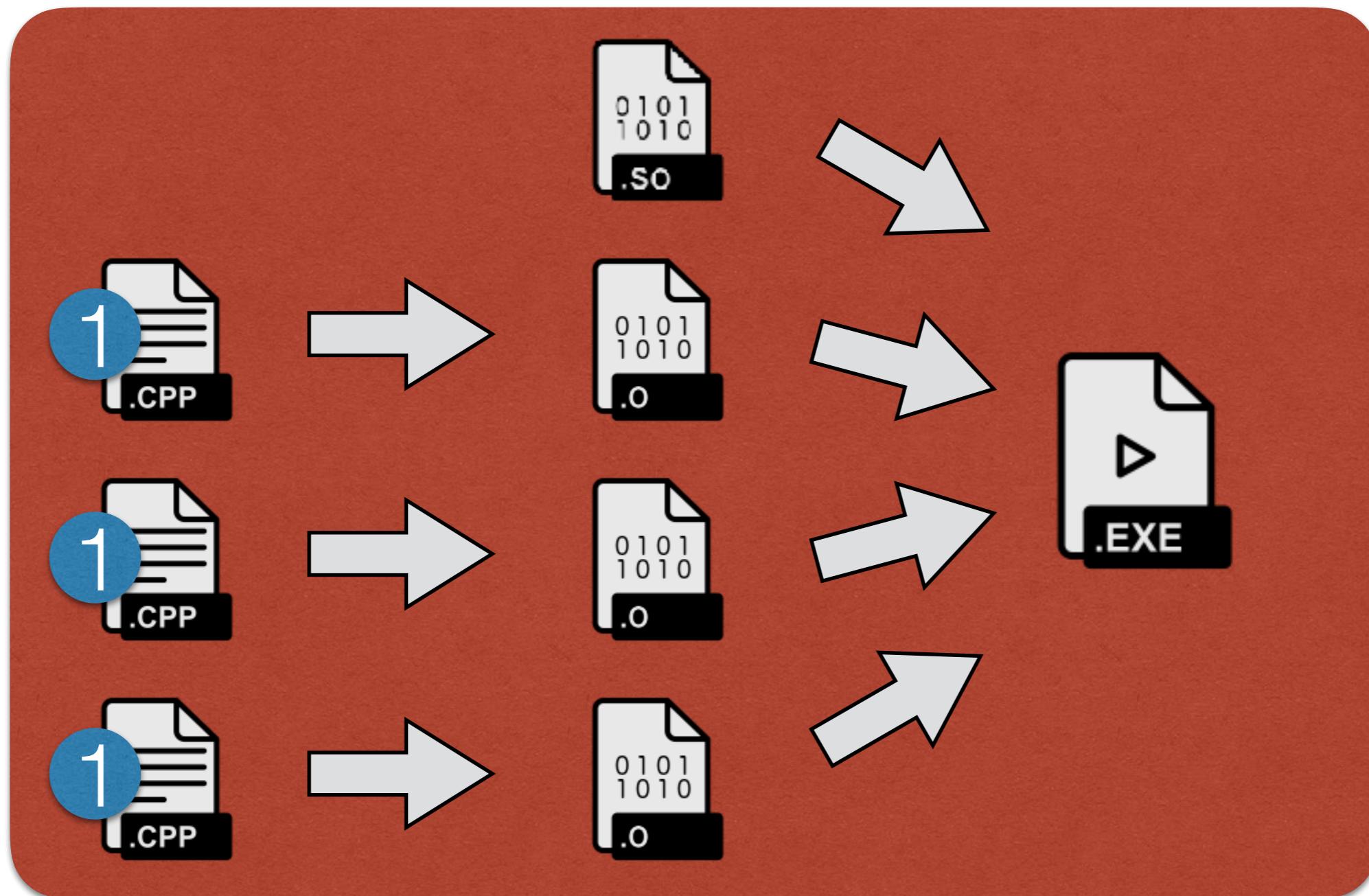


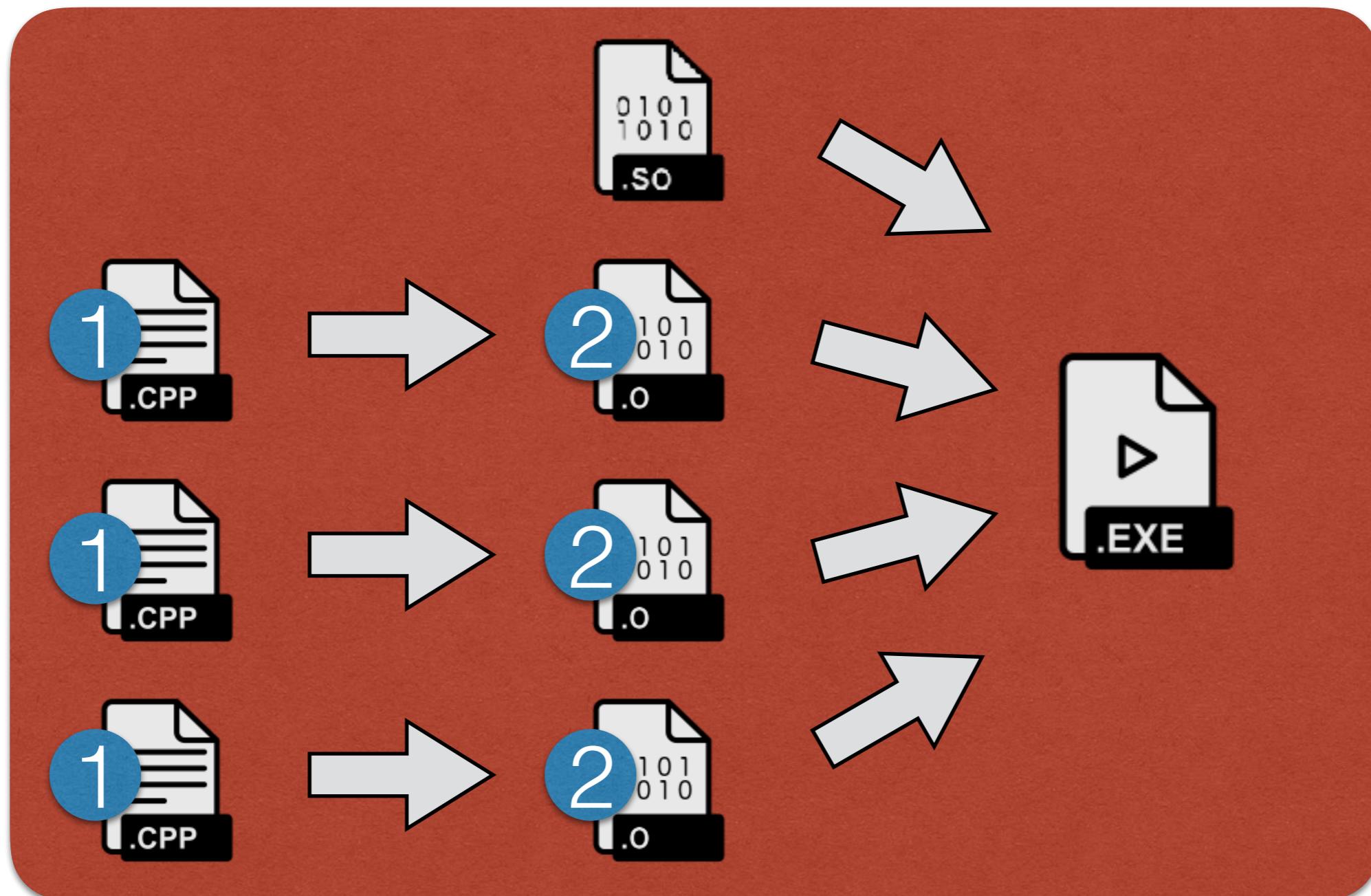


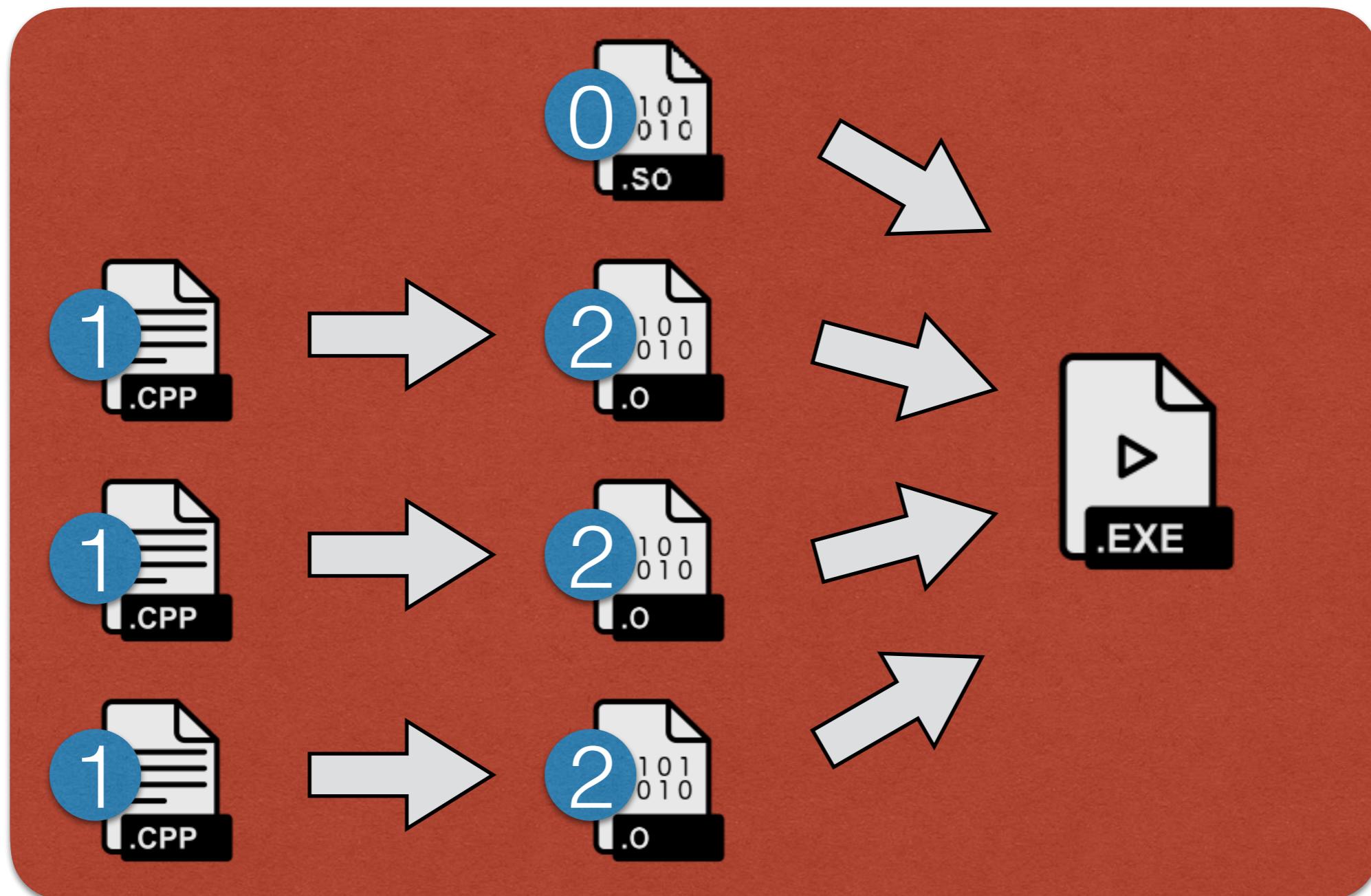


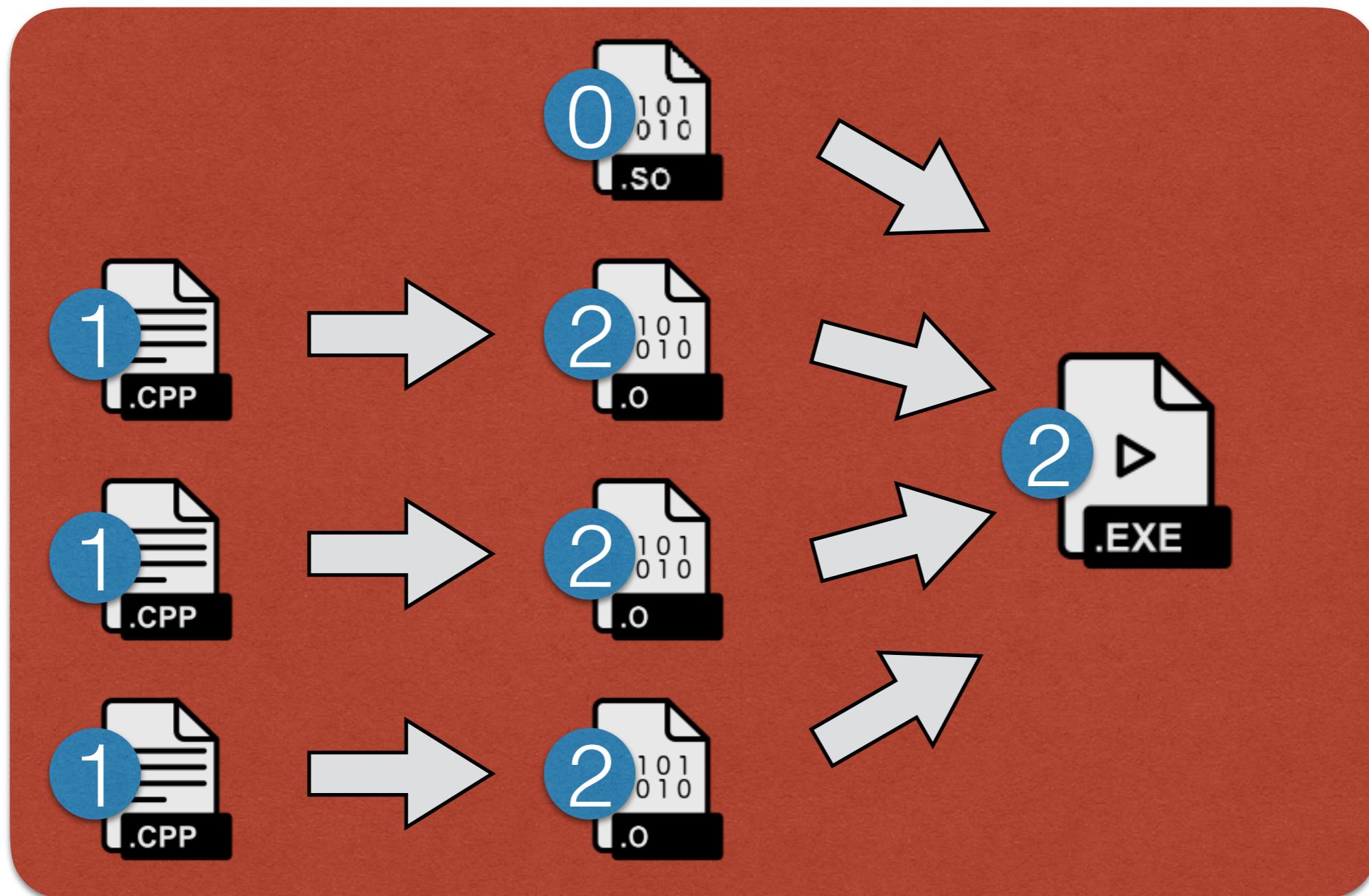


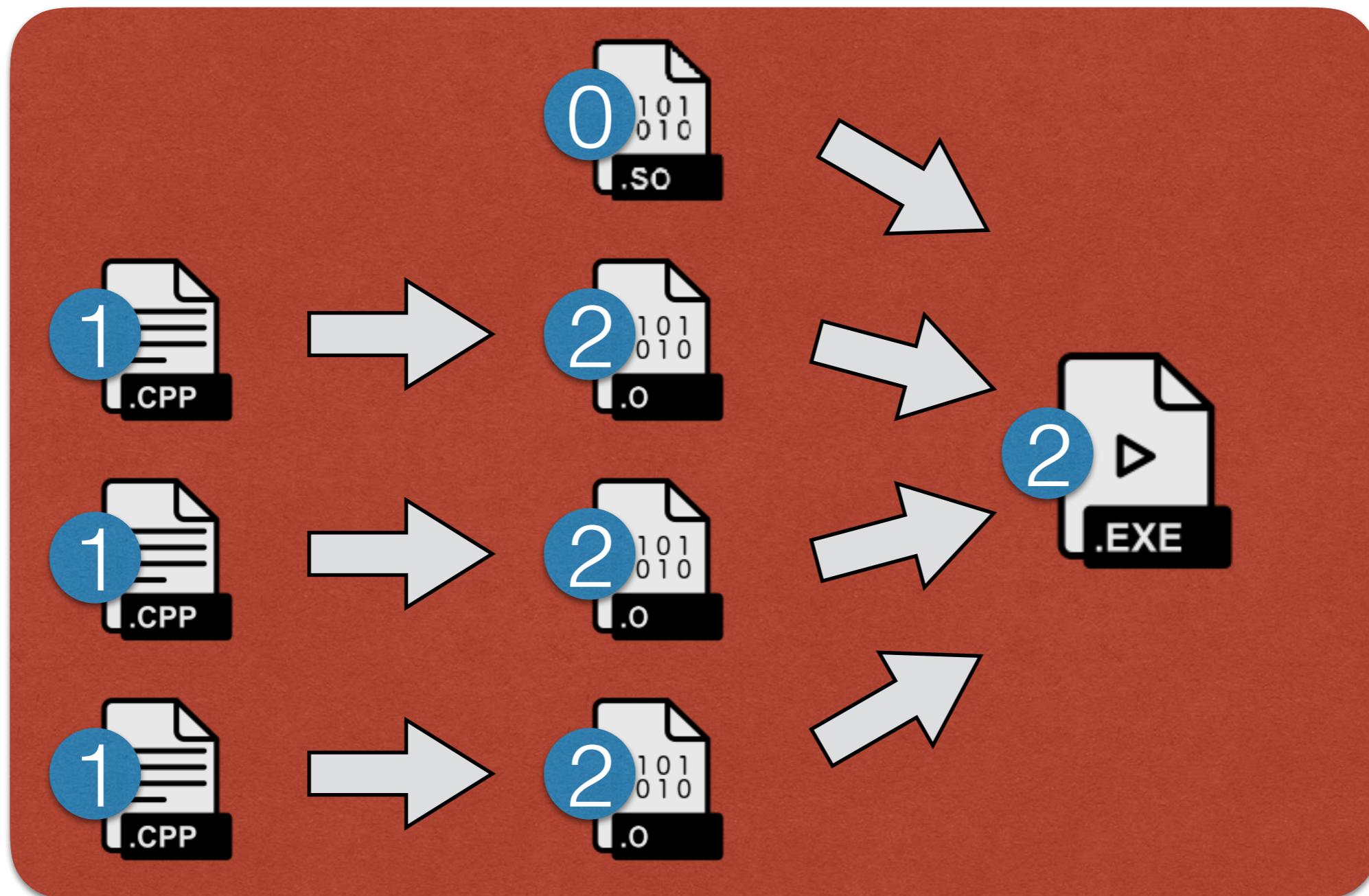


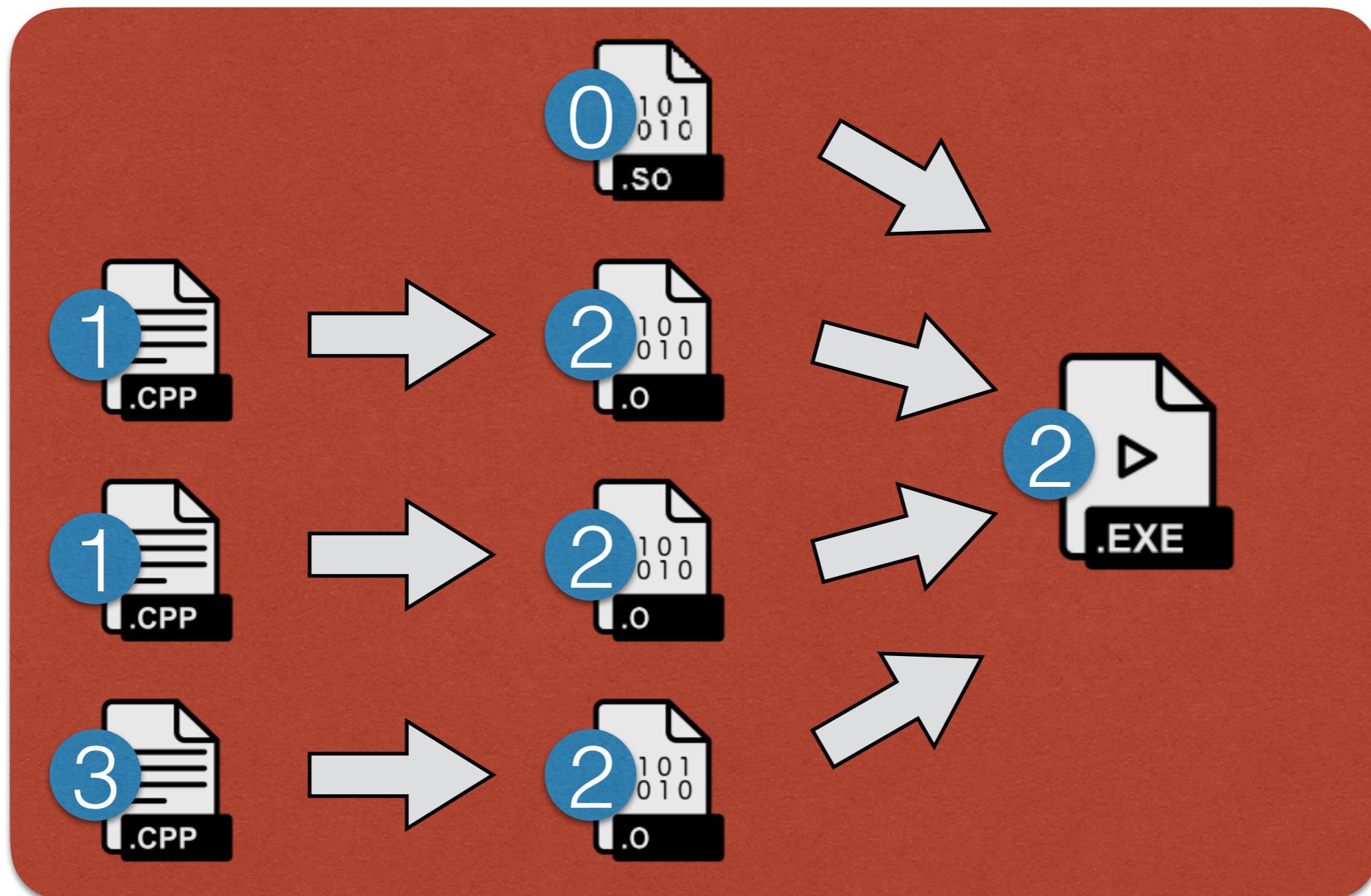


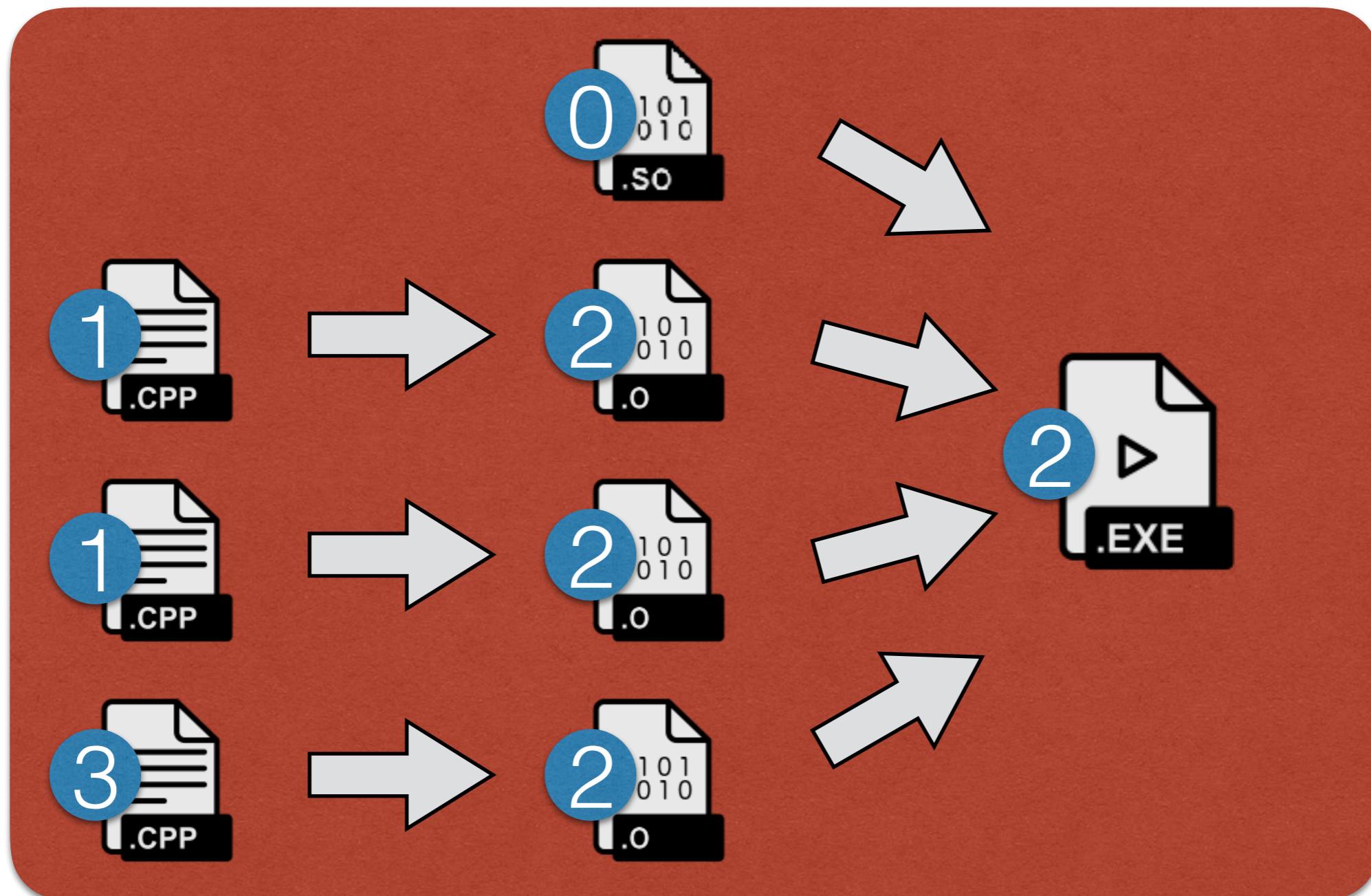


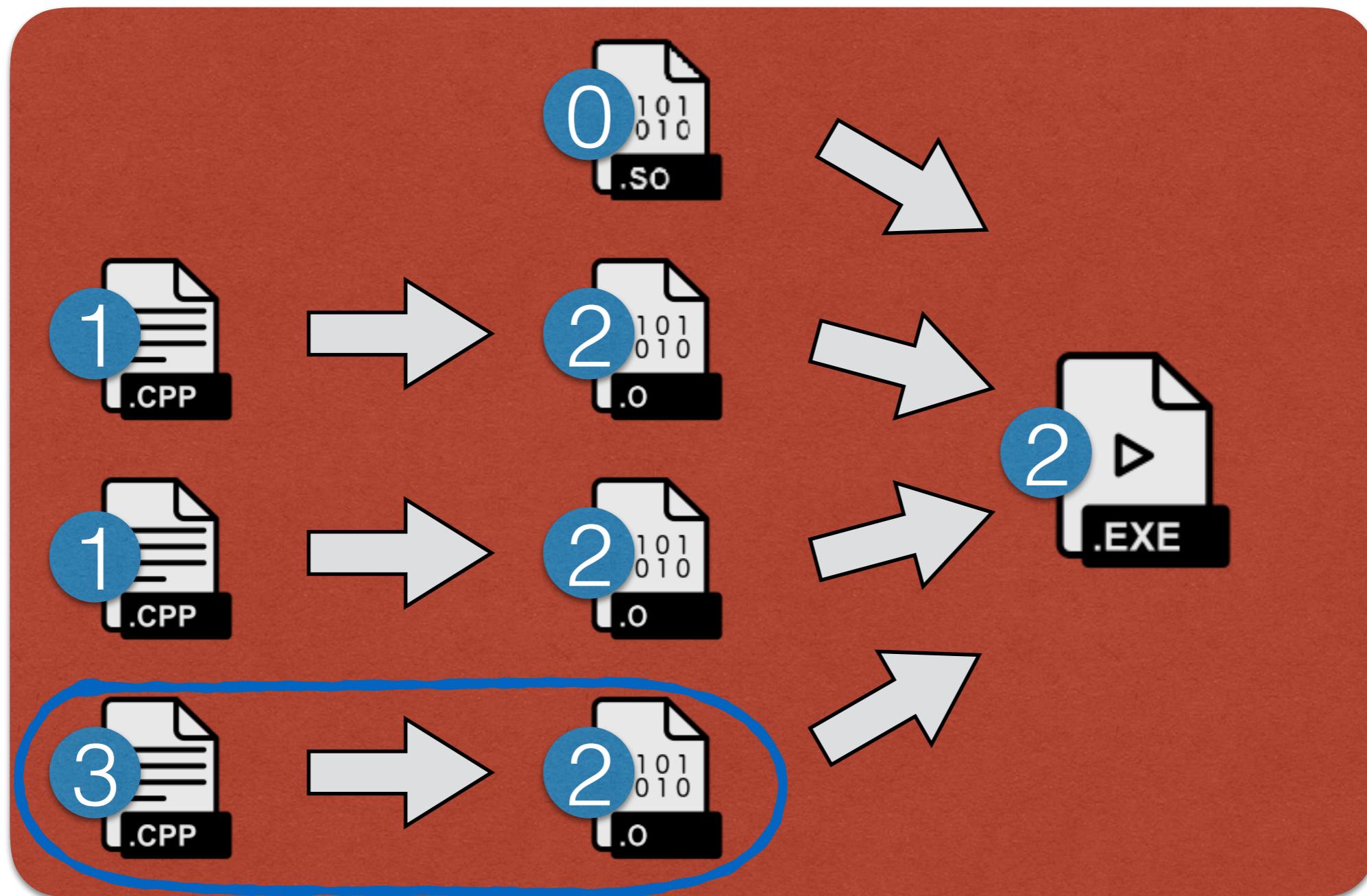


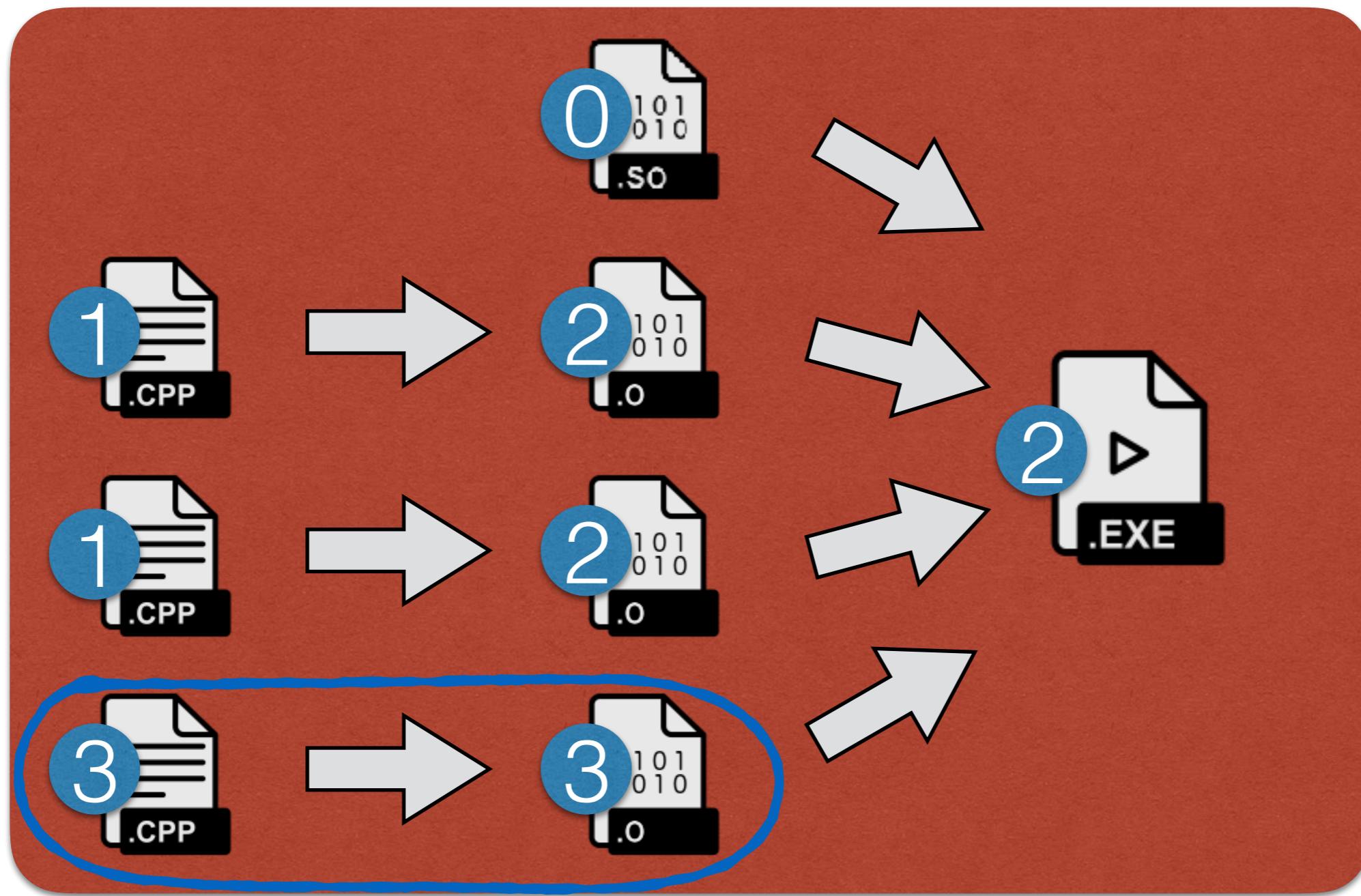


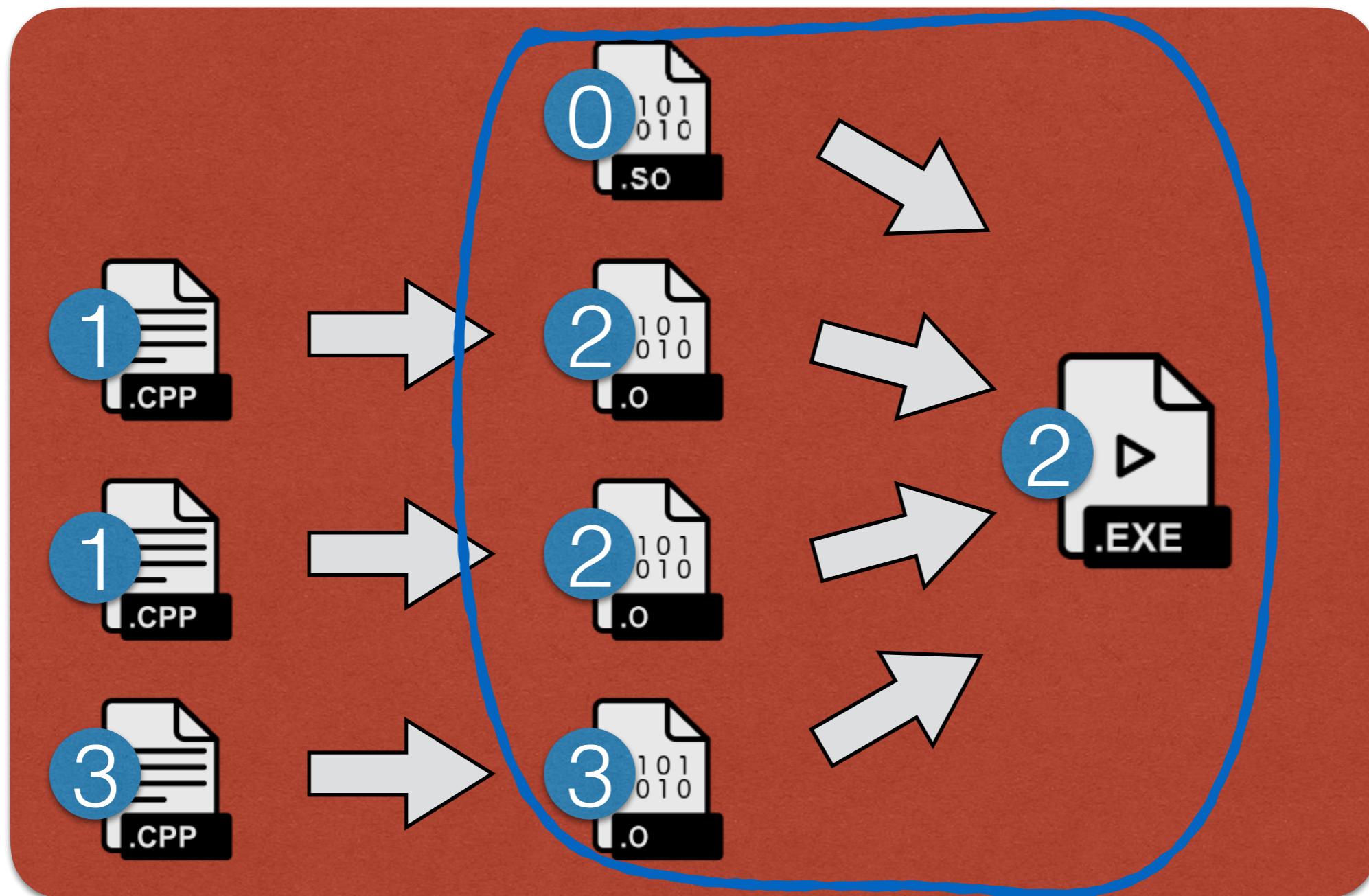


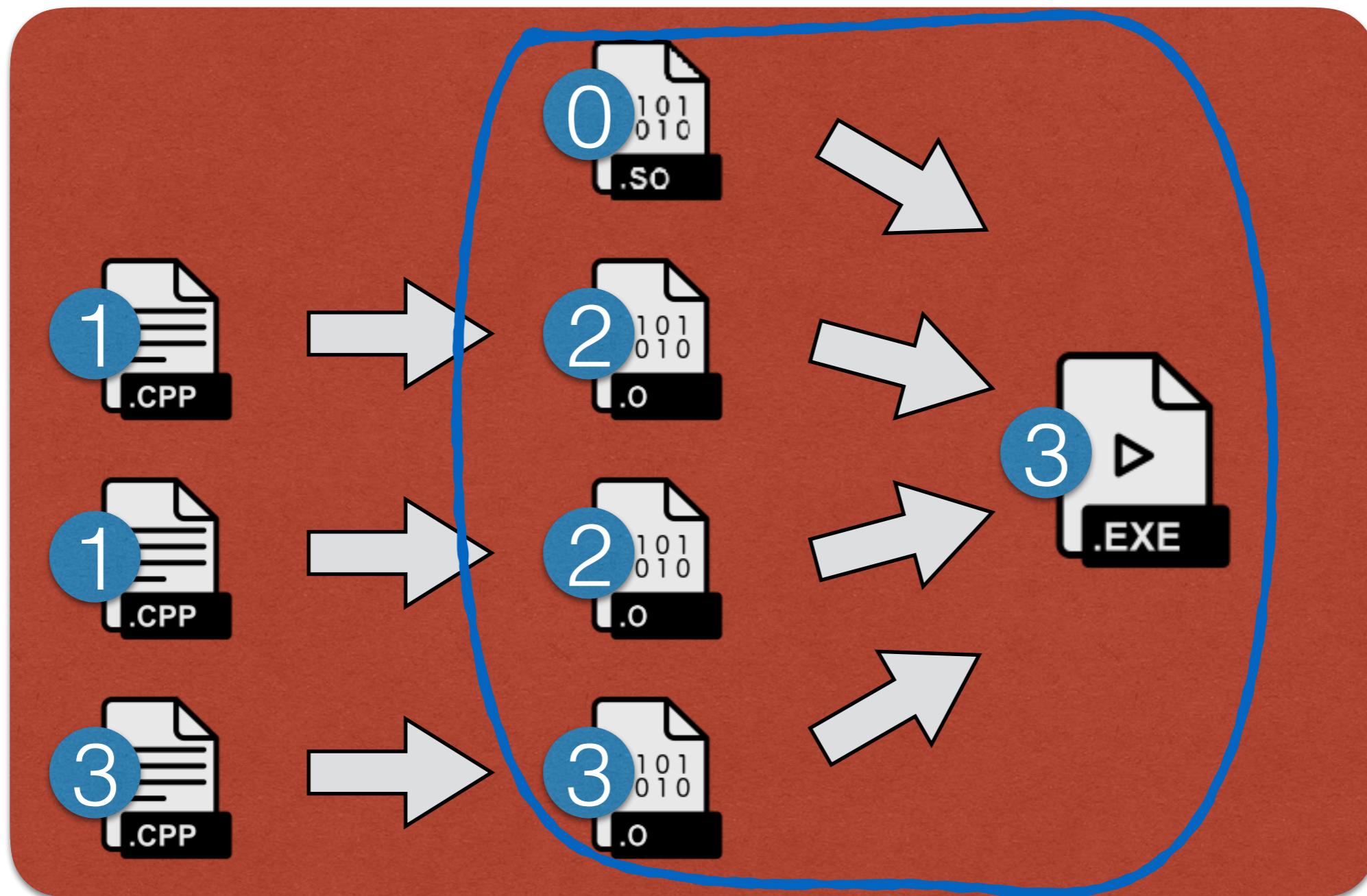


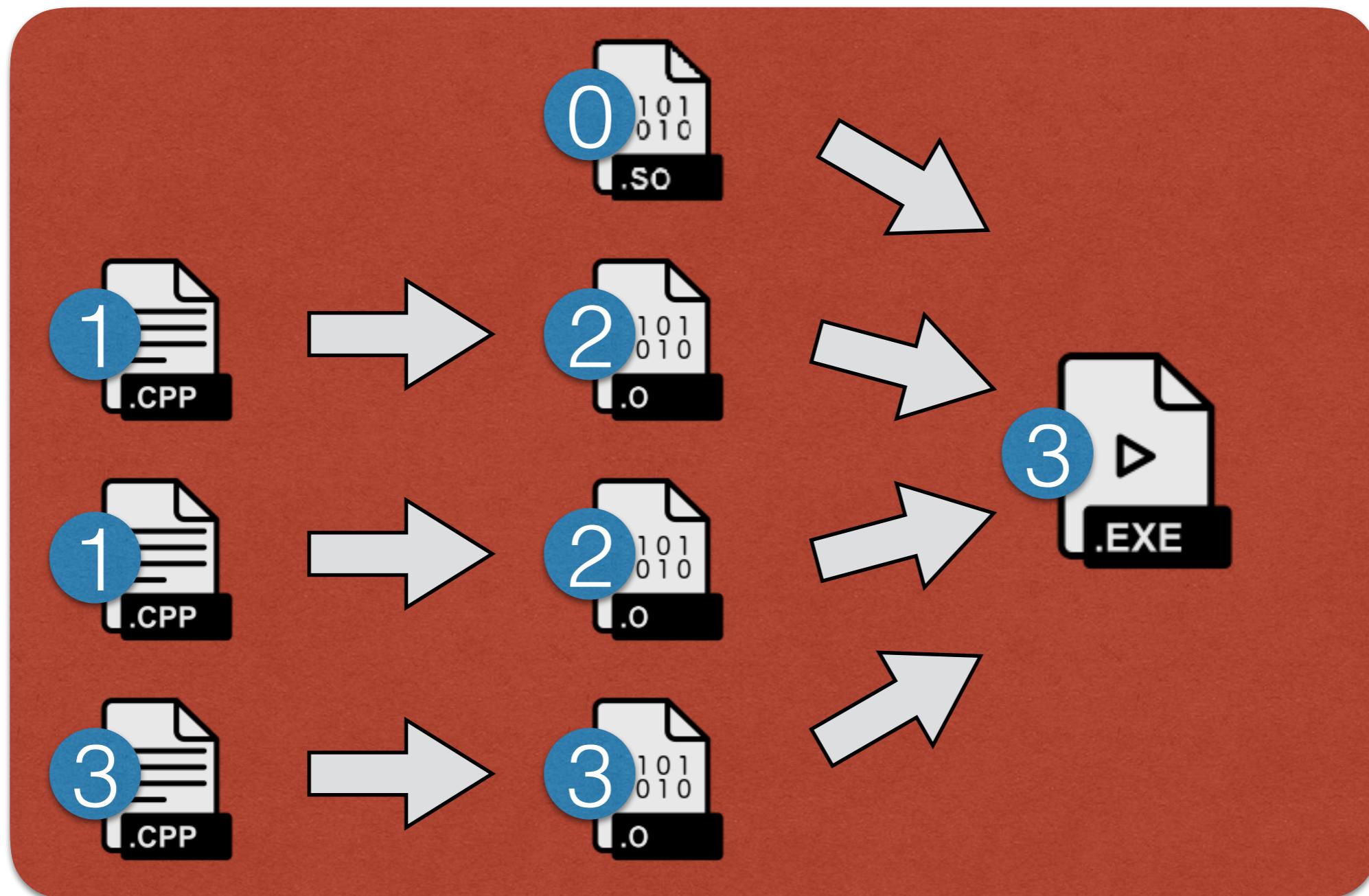


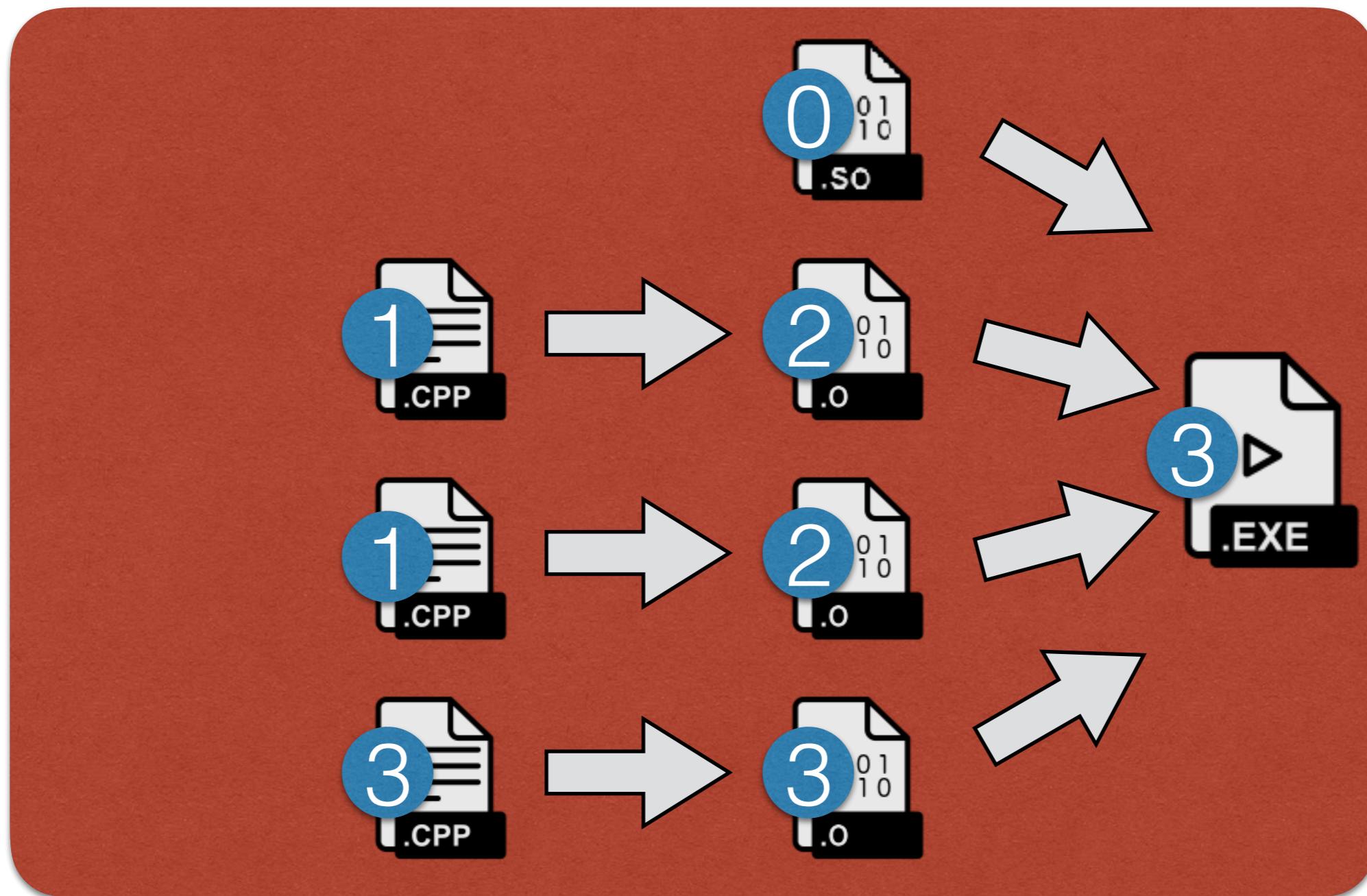


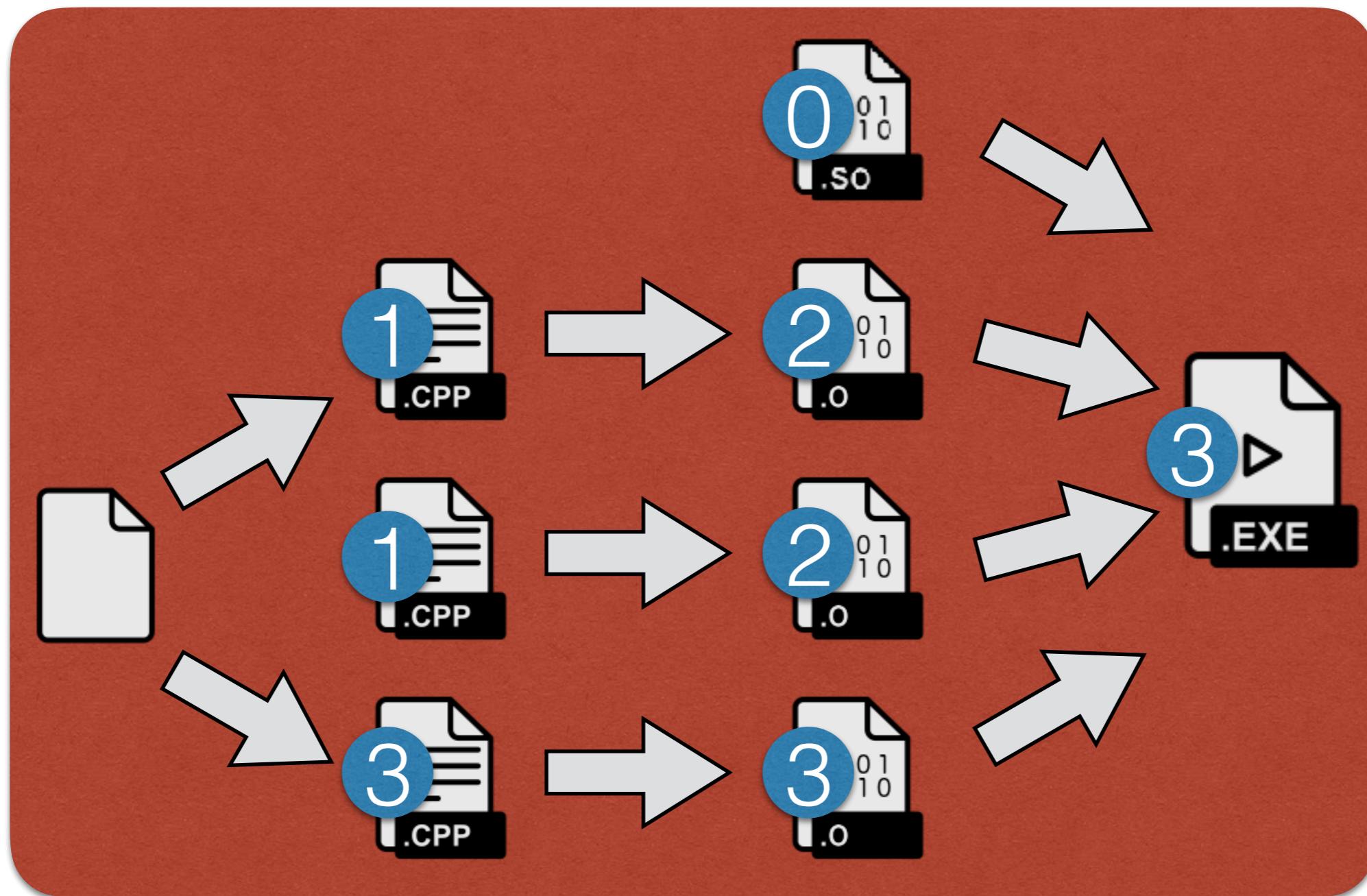


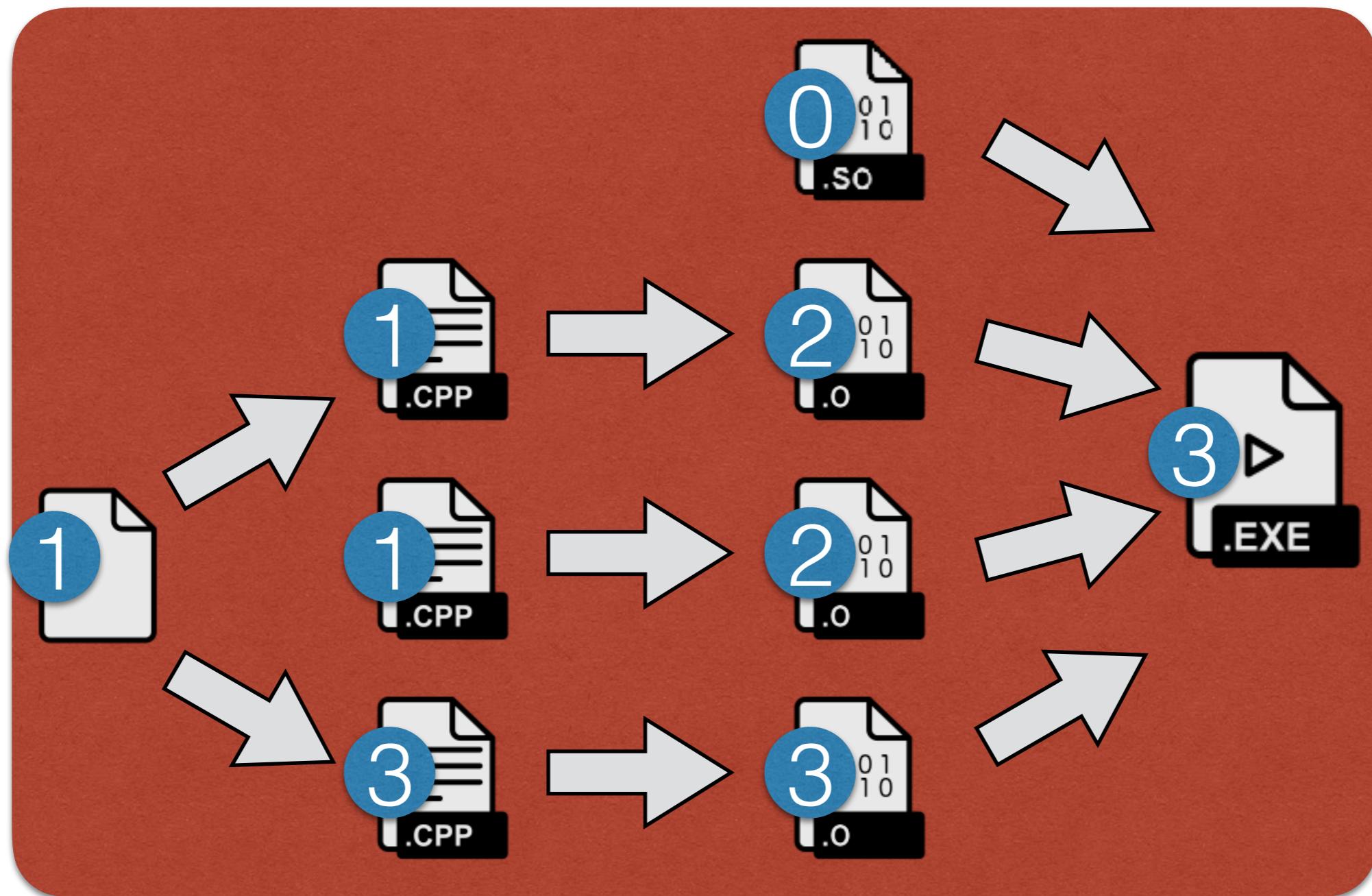


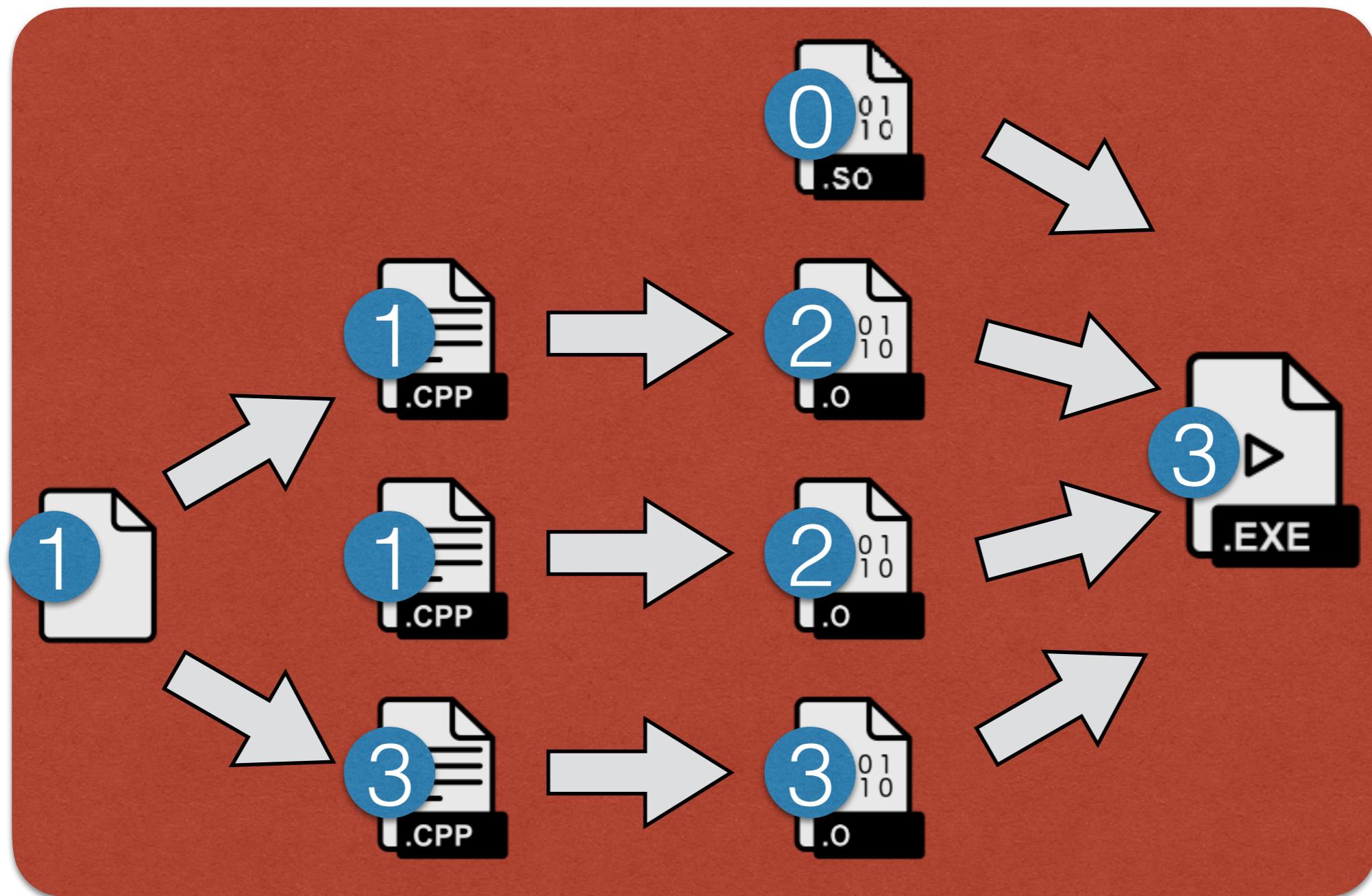


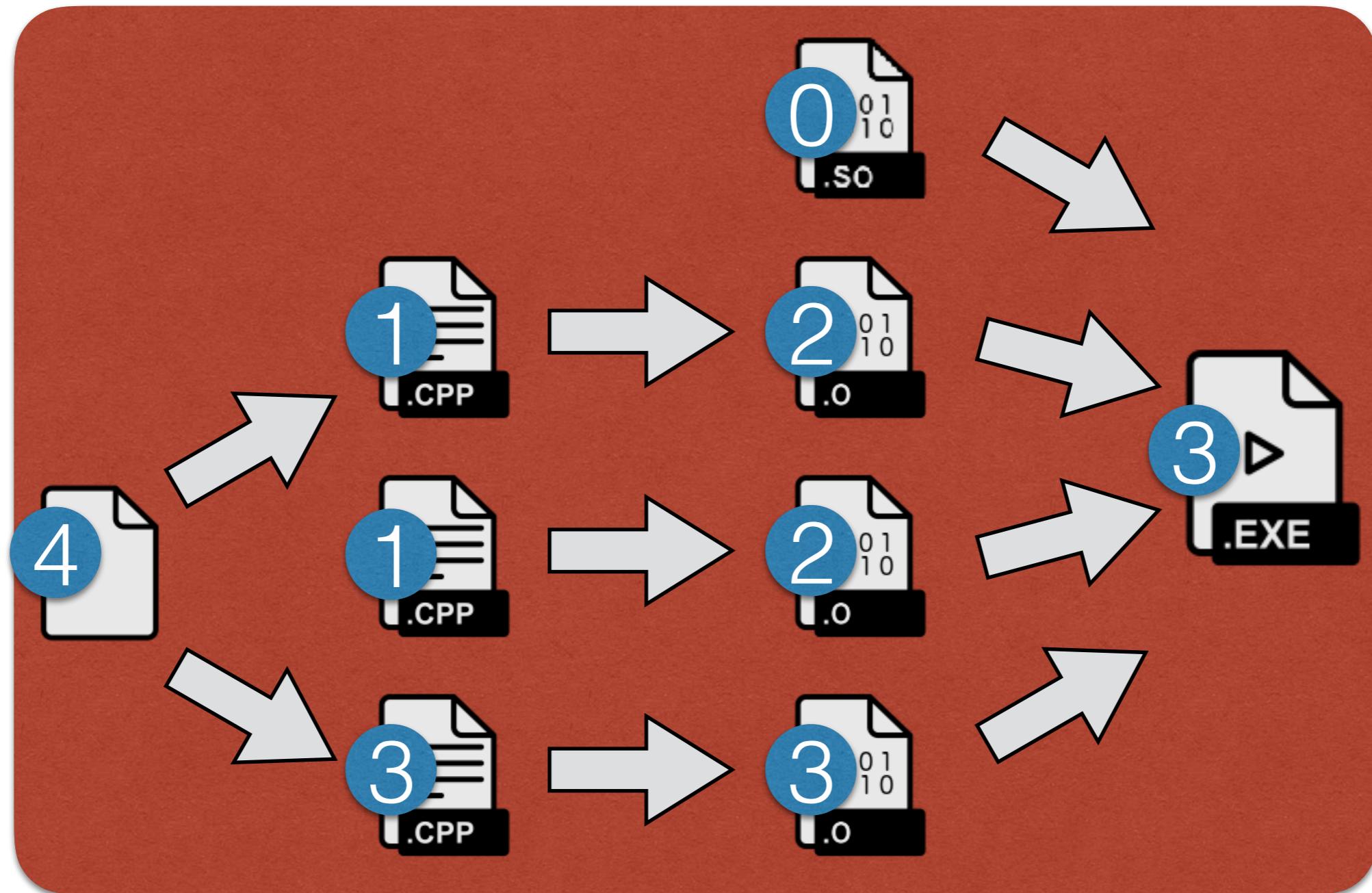


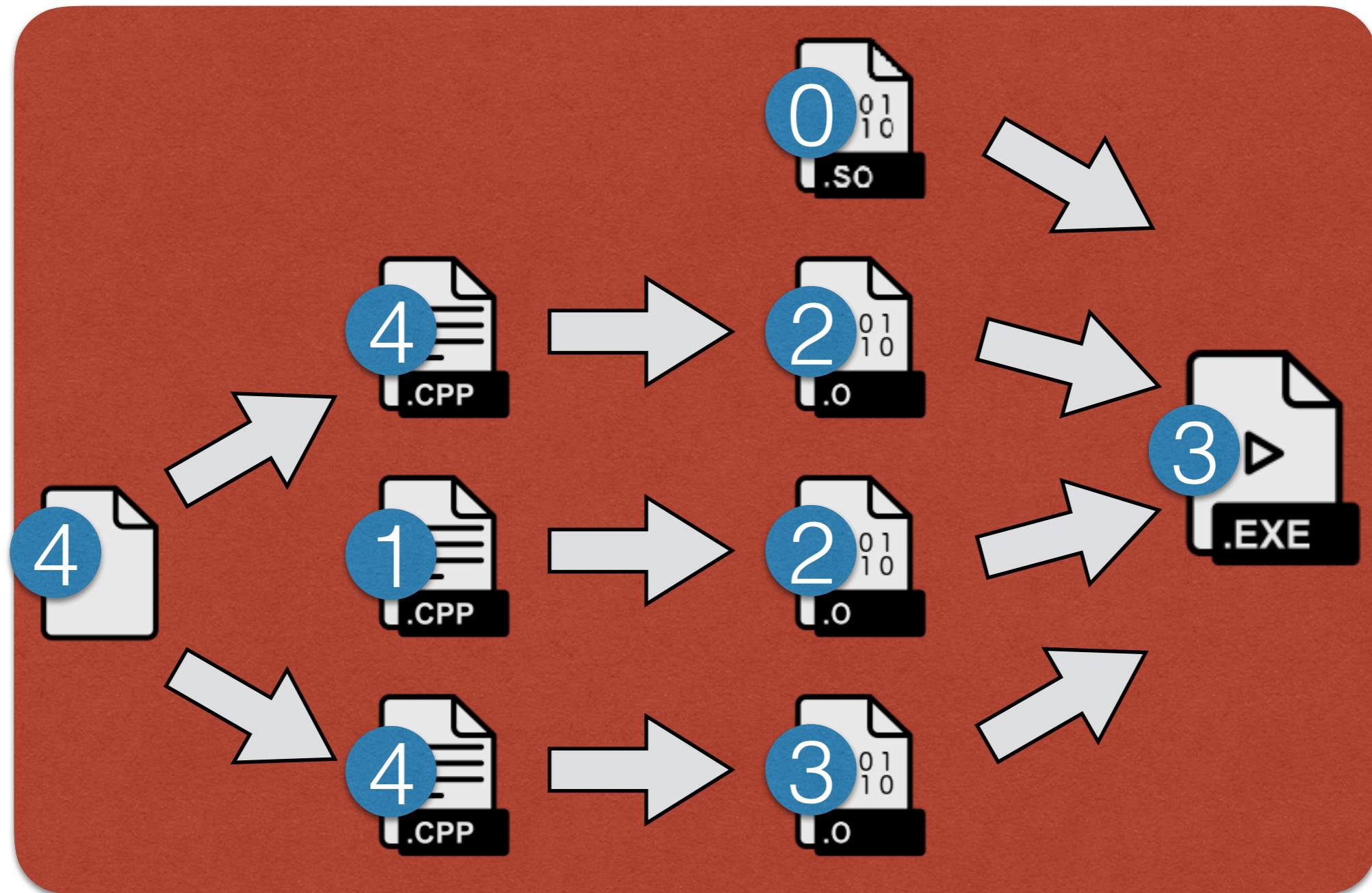


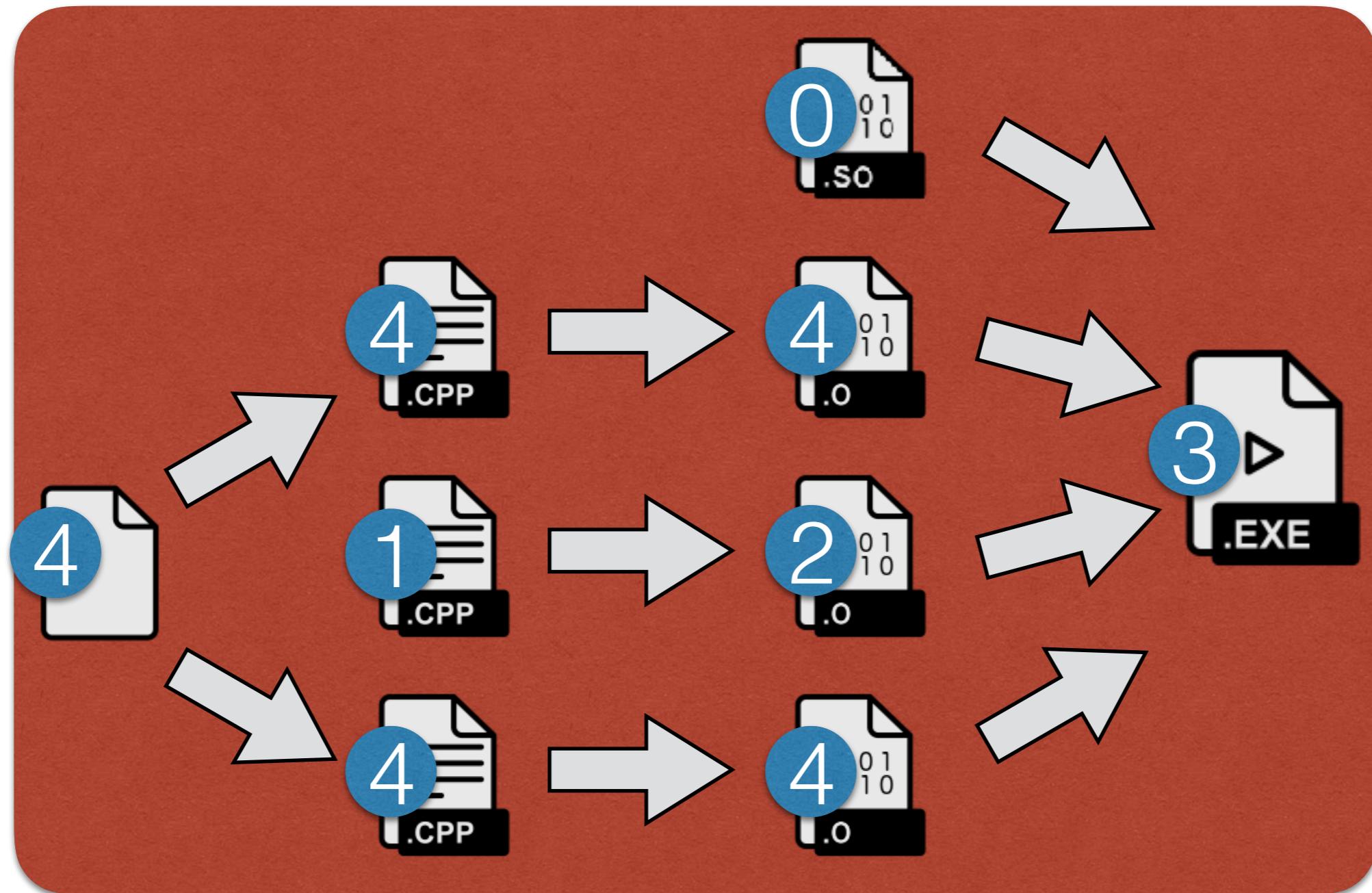


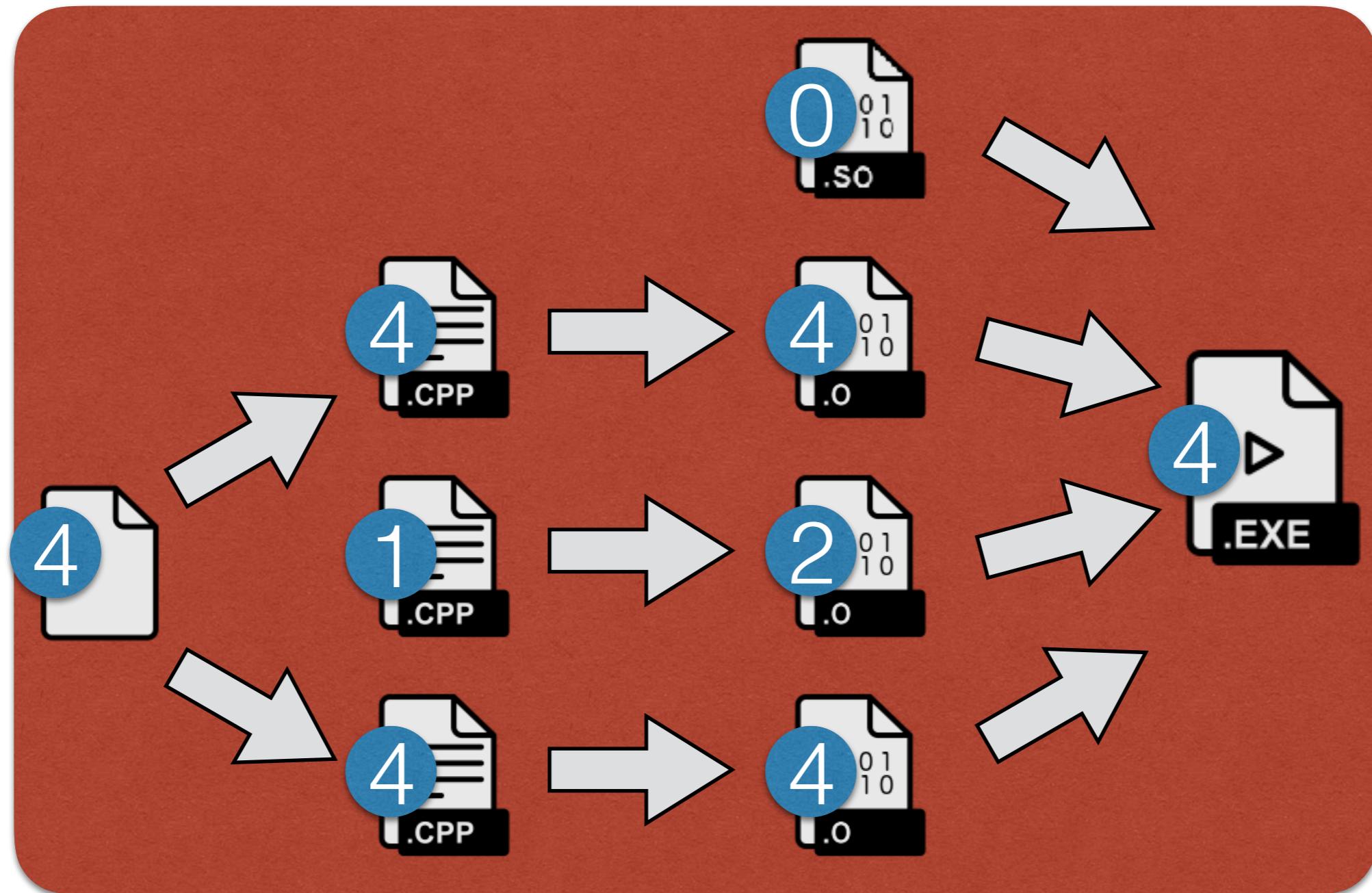


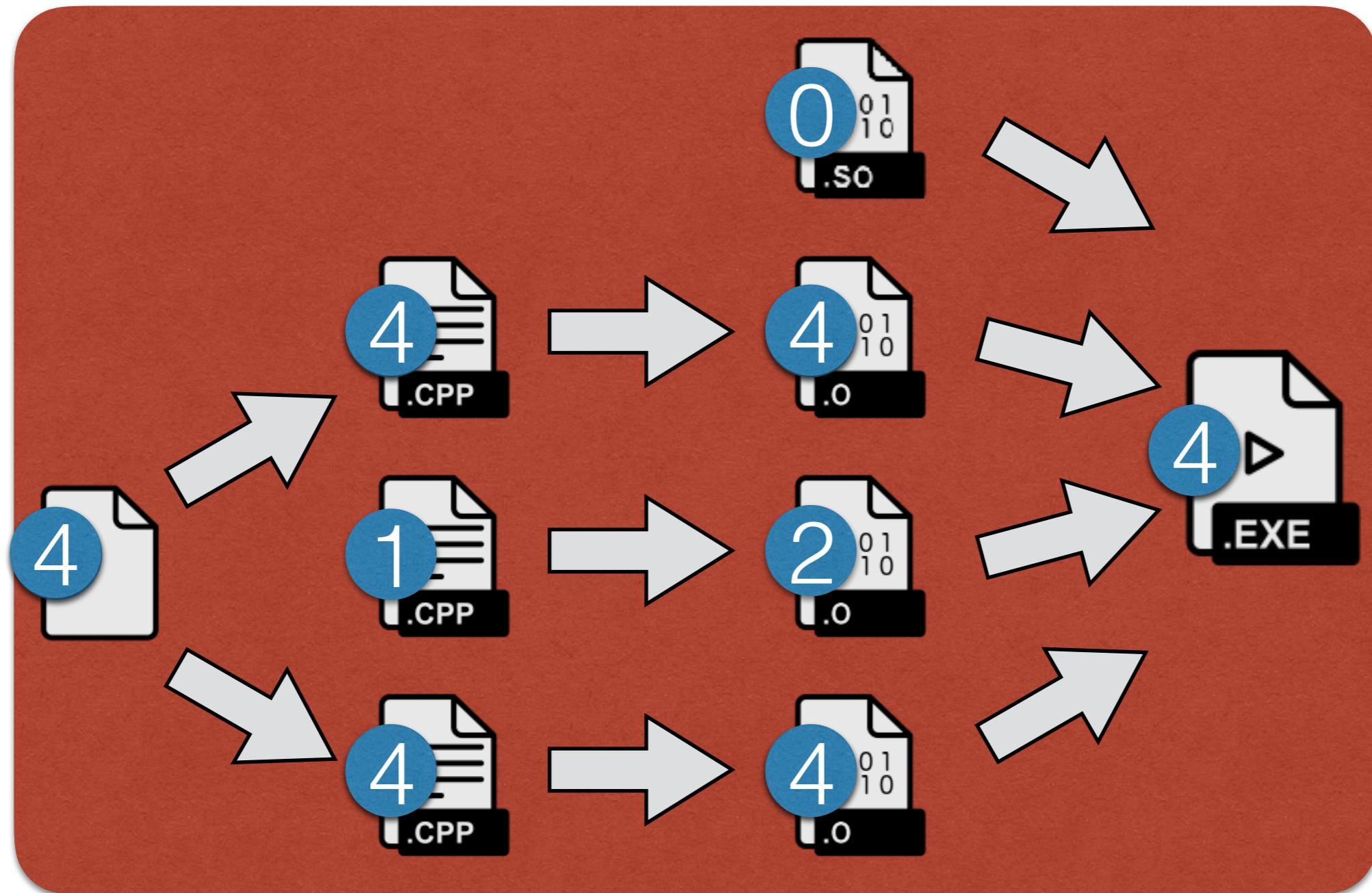




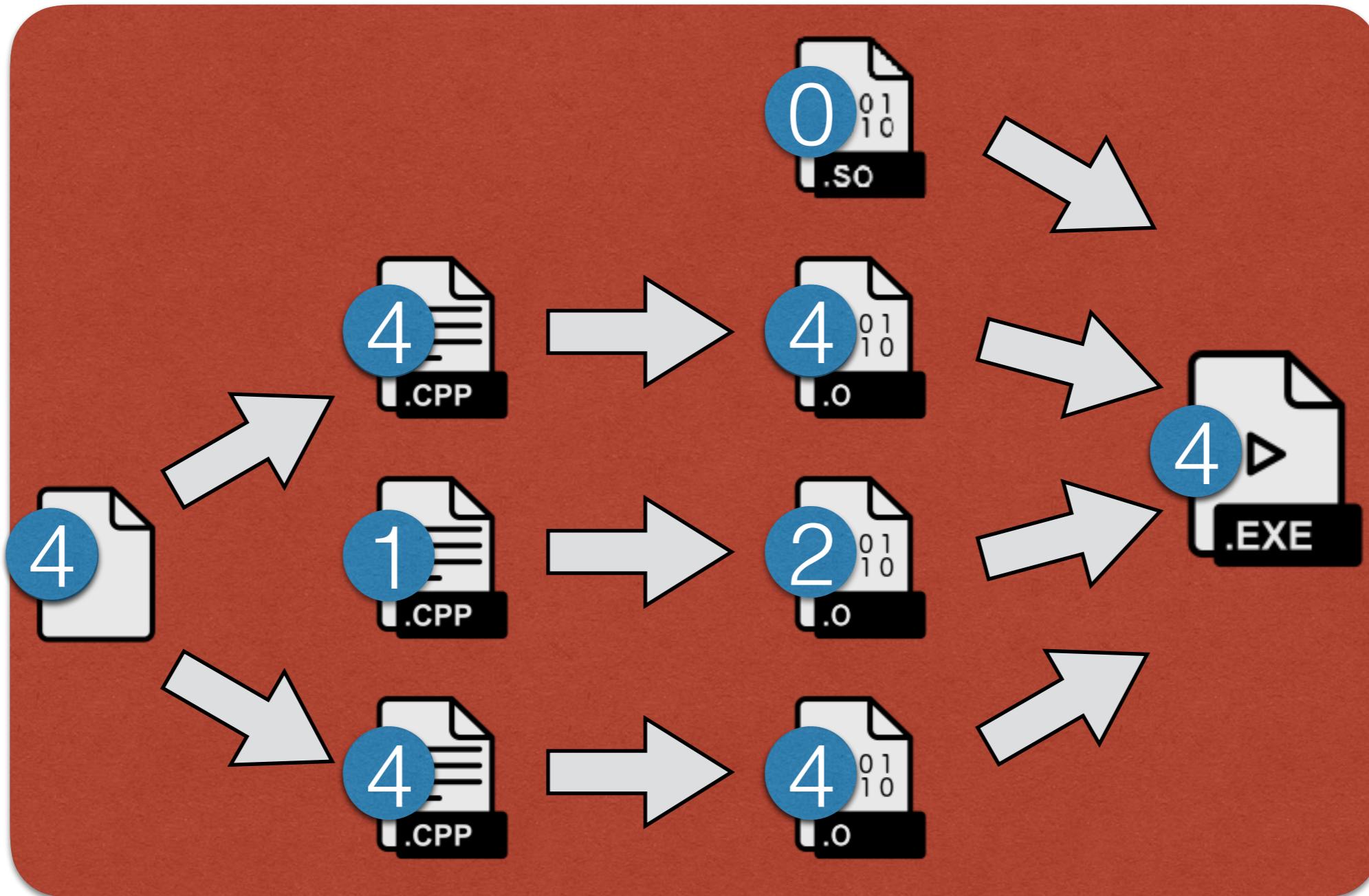




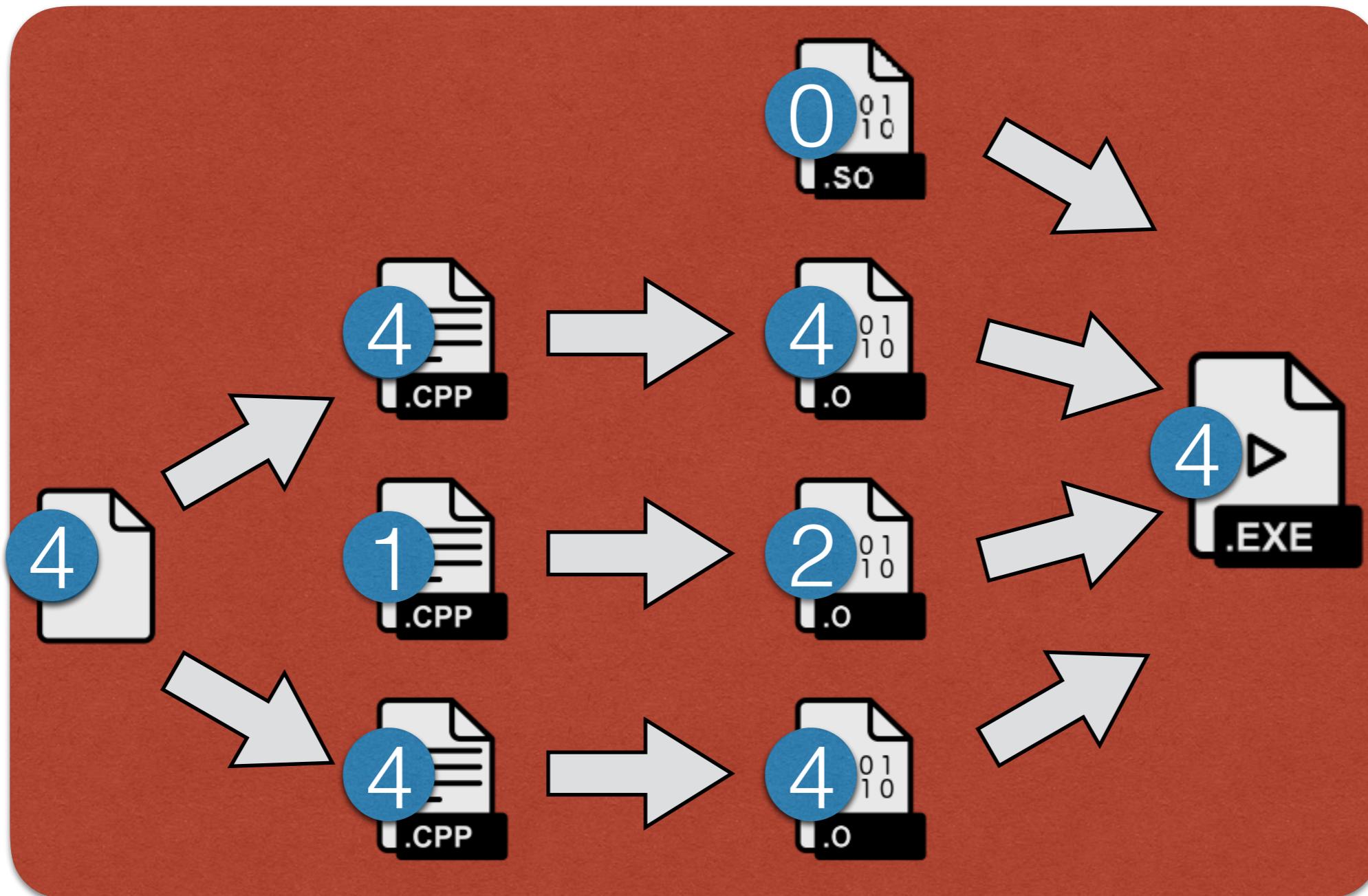




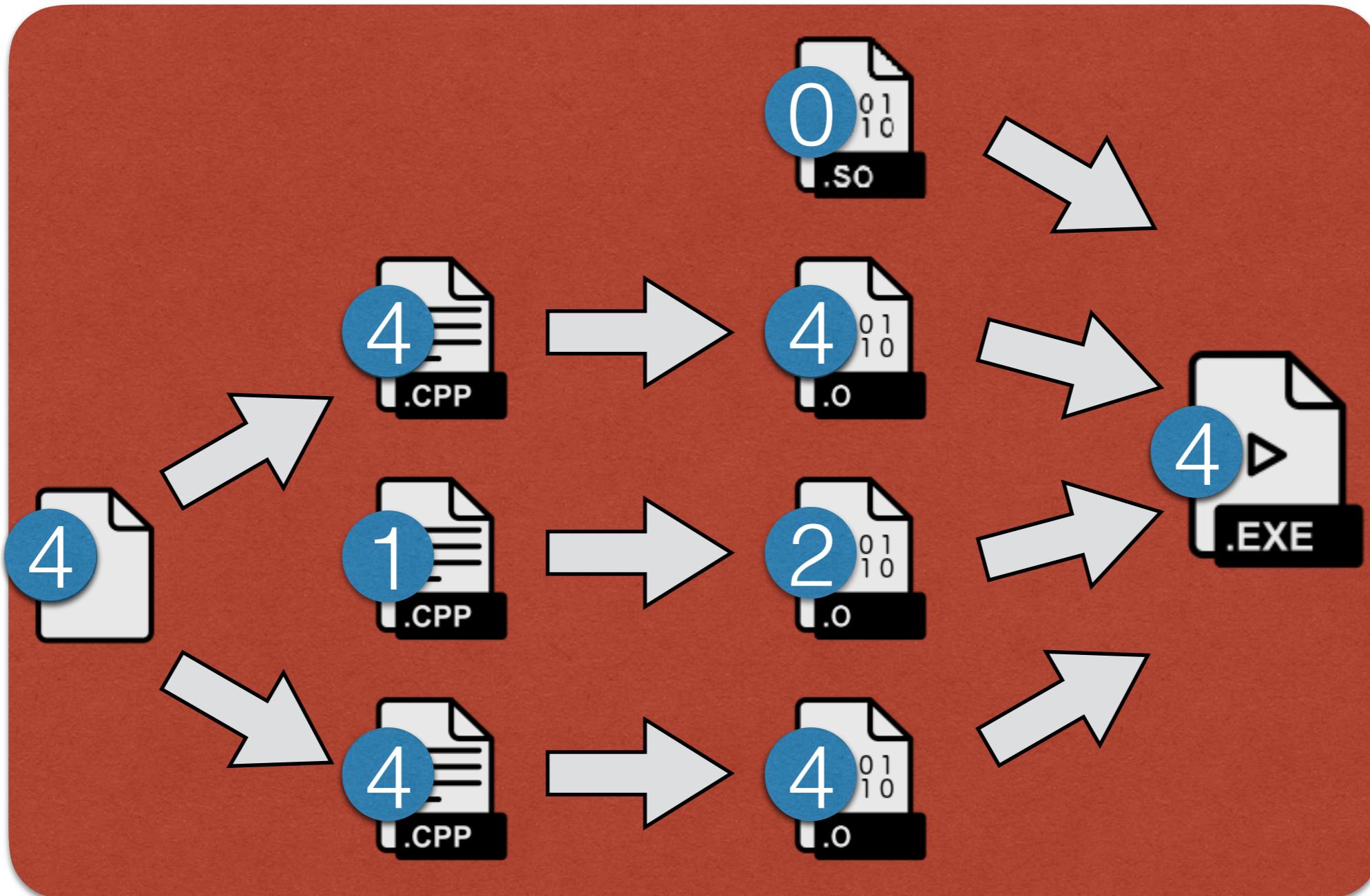
```
clang++ -c -MD main.cpp -o main.o
```



```
clang++ -c -MD main.cpp -o main.o
```

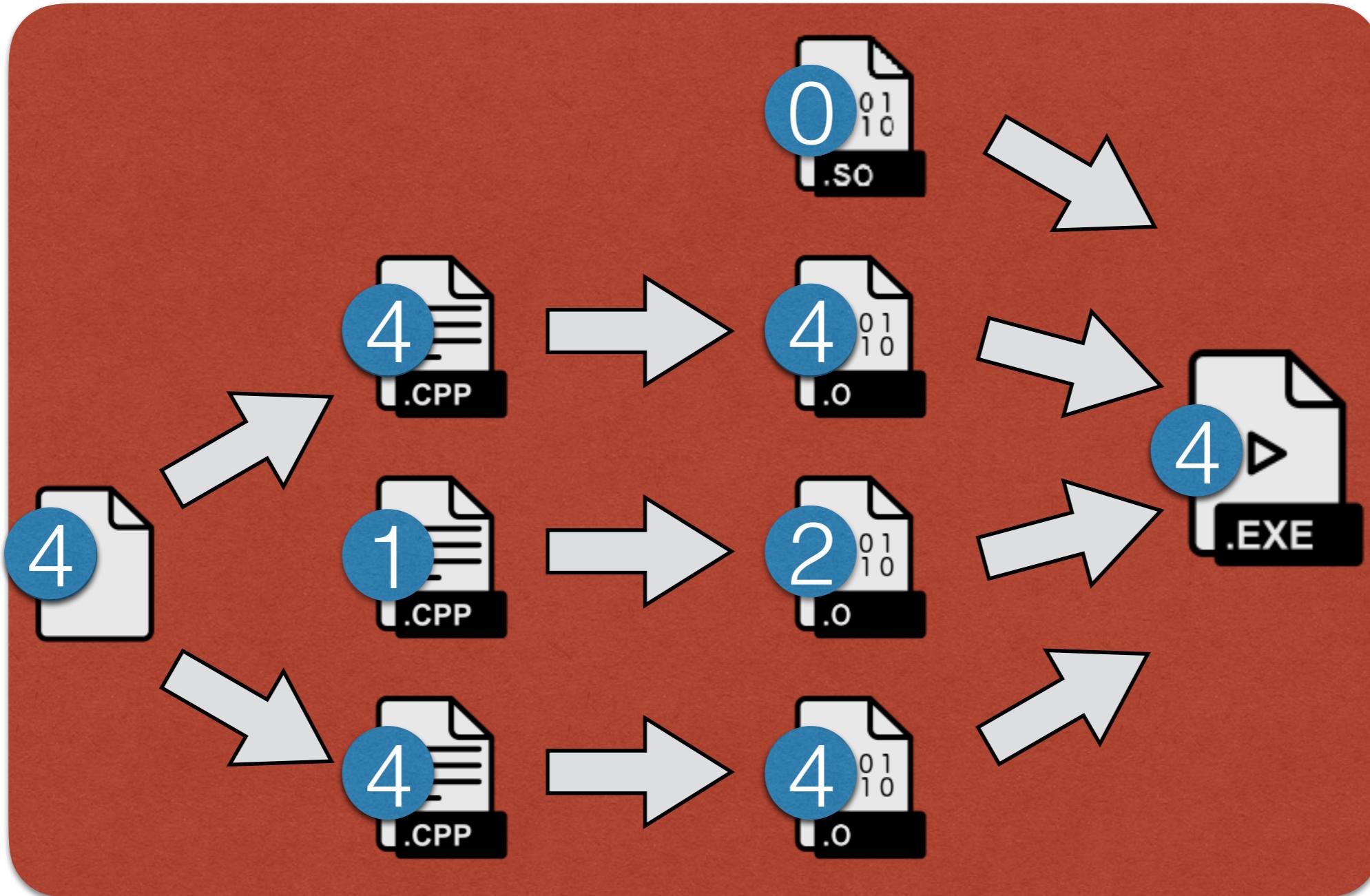


```
clang++ -c -MD main.cpp -o main.o
```

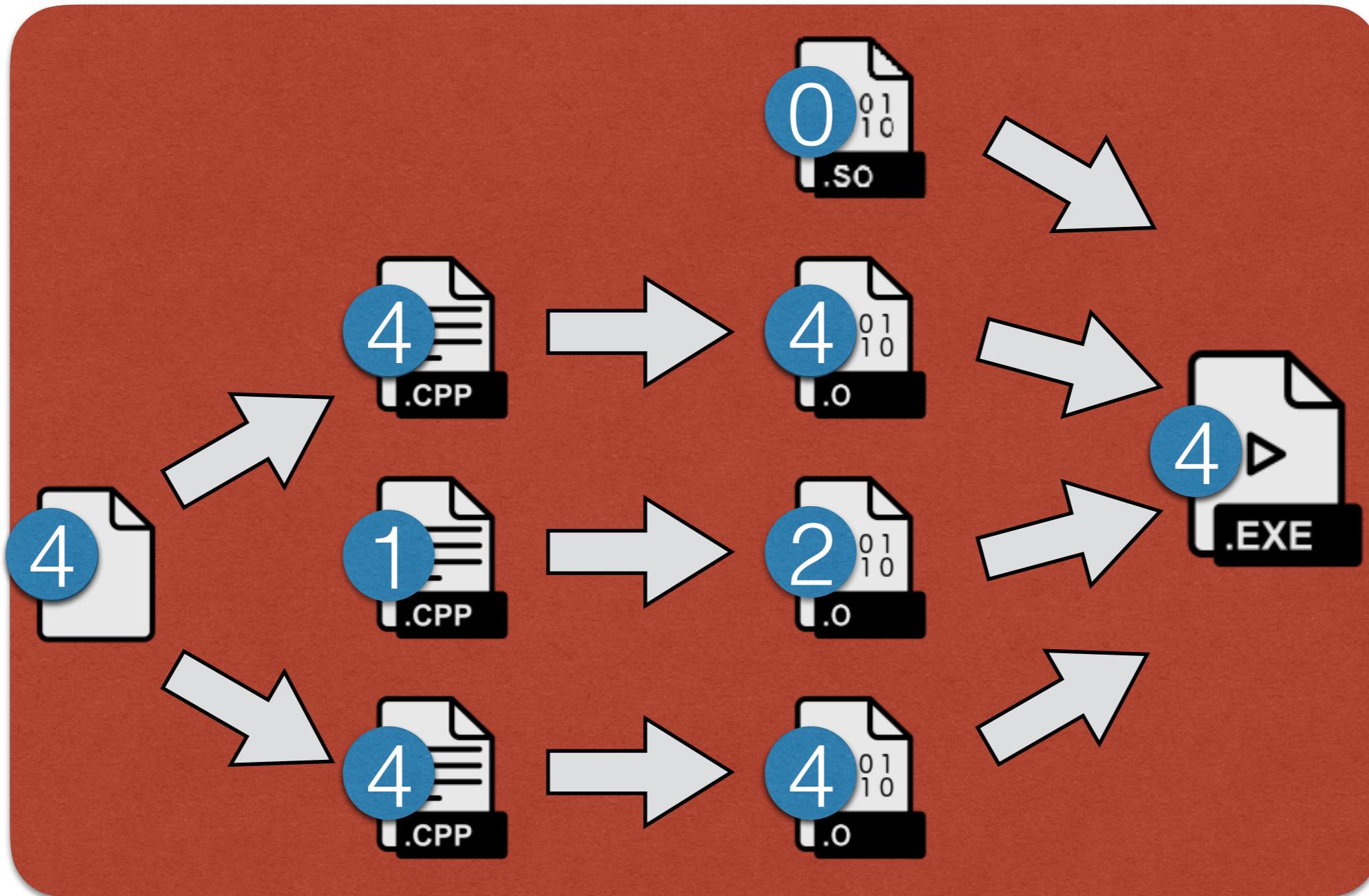


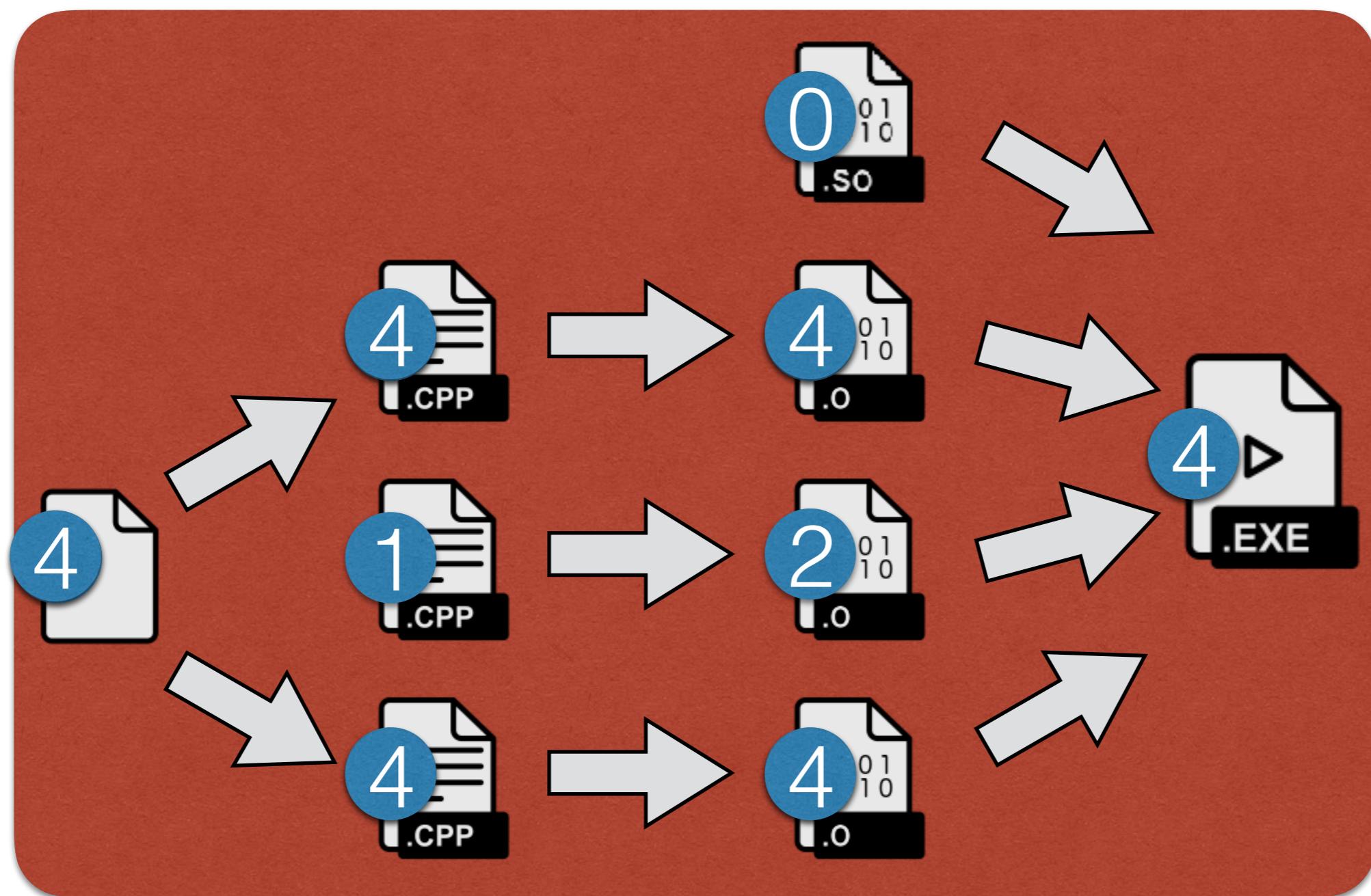
```
clang++ -c -MD main.cpp -o main.o
```

```
main.o: main.cpp CloudBasedSuperAI.h
```

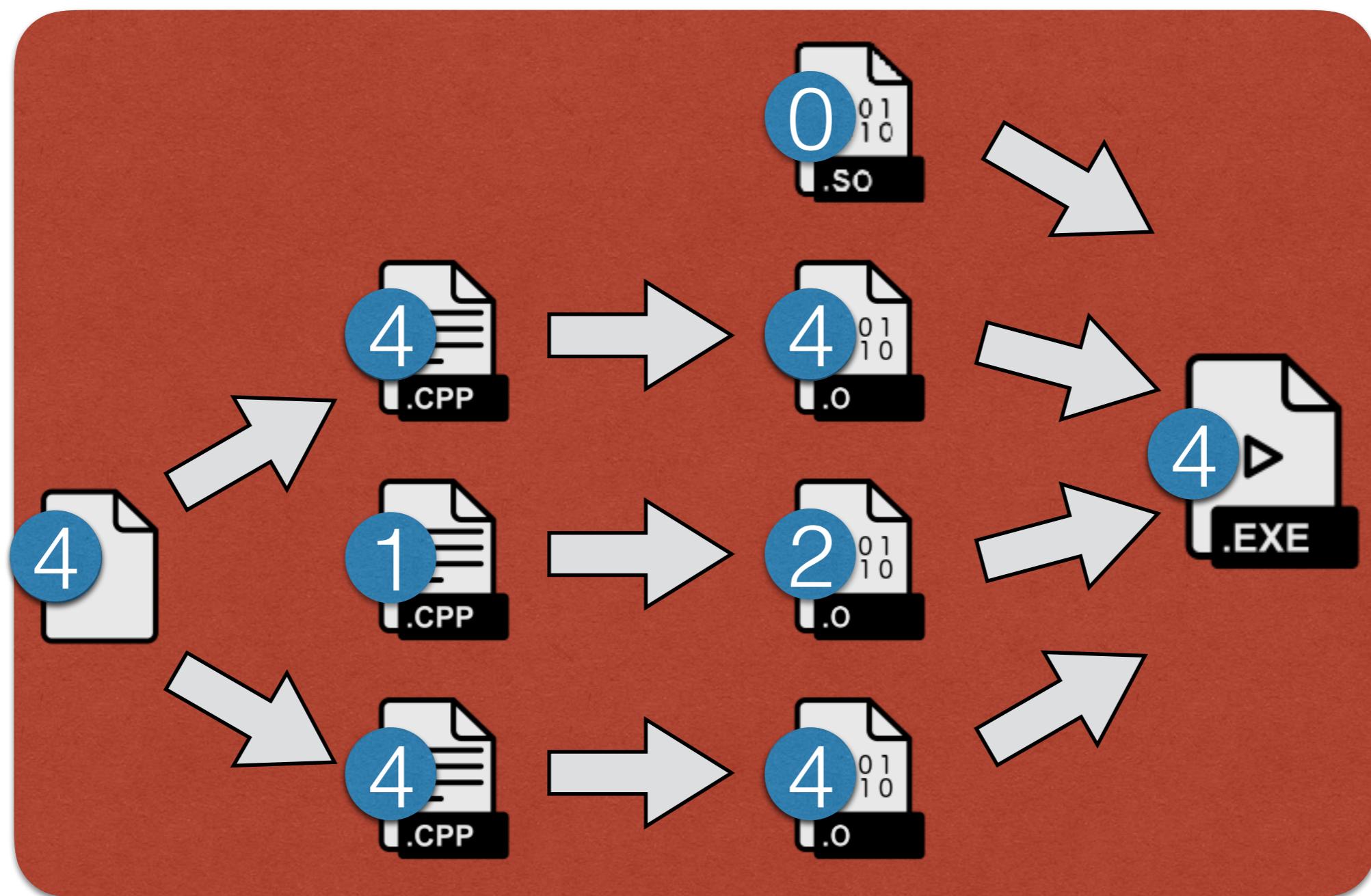


```
clang++ -c -MD main.cpp -o main.o
```









Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

```
otool -L main
```

```
main:
```

```
  /usr/lib/libc++.1.dylib (compatibility version 1.0.0,  
  current version 307.5.0)  
  /usr/lib/libSystem.B.dylib (compatibility version 1.0.0,  
  current version 1238.50.2)
```

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

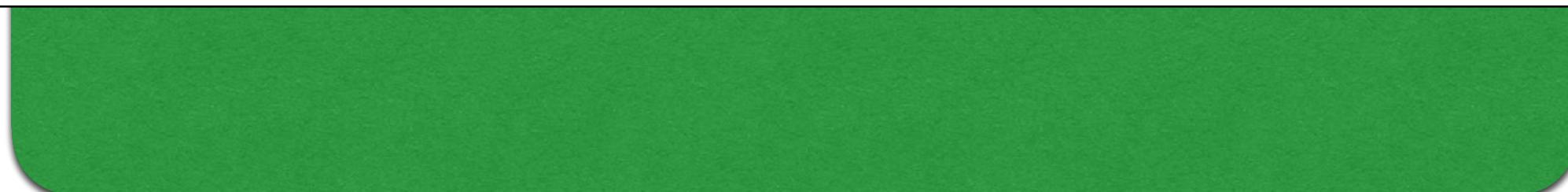
- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

DYLD_PRINT_LIBRARIES=1 ./main

```
dyld: loaded: main
dyld: loaded: /usr/lib/libc++.1.dylib
dyld: loaded: /usr/lib/libSystem.B.dylib
dyld: loaded: /usr/lib/libc++abi.dylib
dyld: loaded: /usr/lib/system/libcache.dylib
dyld: loaded: /usr/lib/system/libcommonCrypto.dylib
dyld: loaded: /usr/lib/system/libcompiler_rt.dylib
dyld: loaded: /usr/lib/system/libcopyfile.dylib
dyld: loaded: /usr/lib/system/libcorecrypto.dylib
dyld: loaded: /usr/lib/system/libdispatch.dylib
dyld: loaded: /usr/lib/system/libdyld.dylib
dyld: loaded: /usr/lib/system/libunwind.dylib
dyld: loaded: /usr/lib/system/libxpc.dylib
```



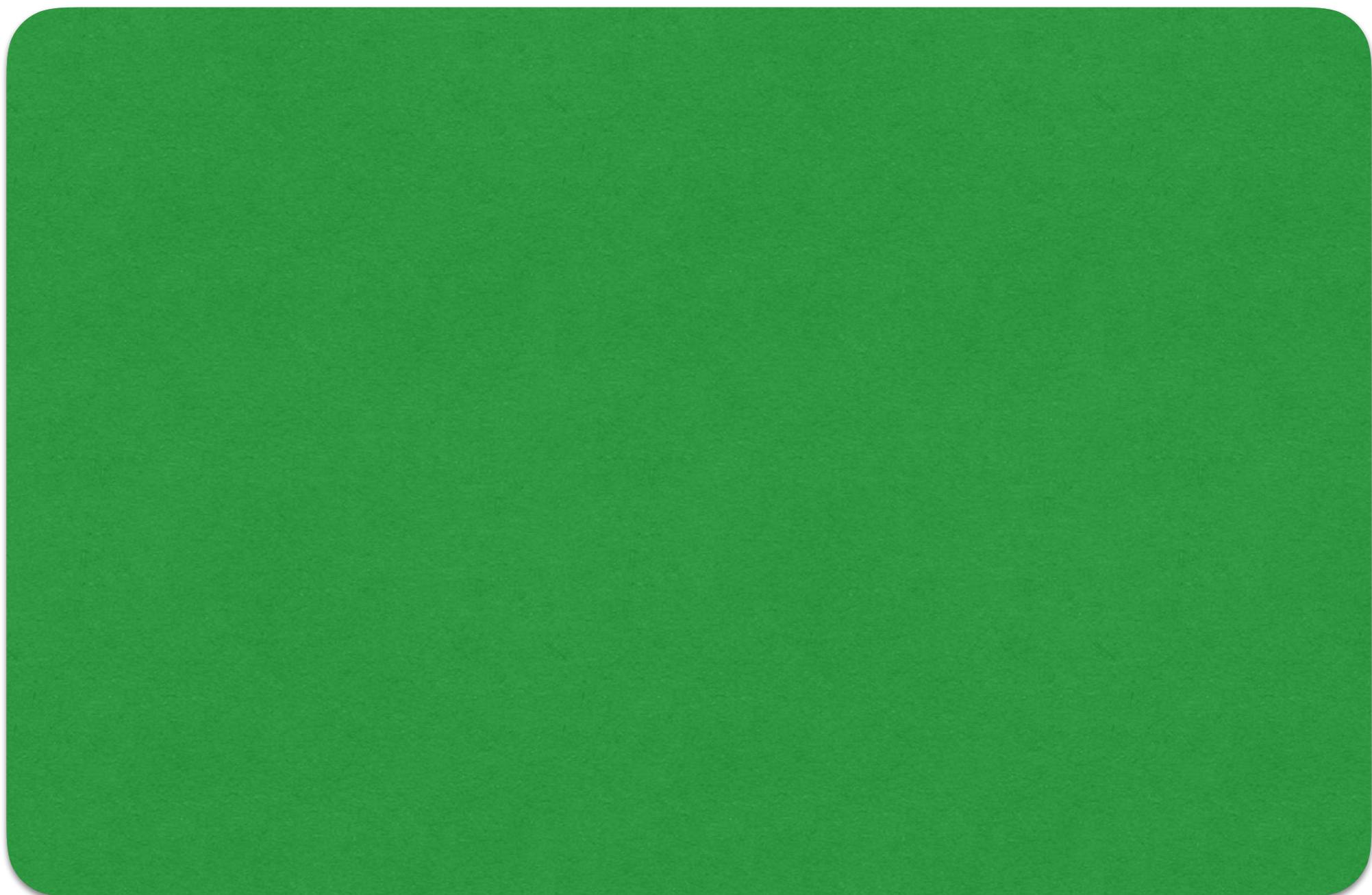
Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

Execute

- Load executable into memory
- Load dynamic linker library into memory
- Execute the dynamic linker

- Call the entry point function
 - `/usr/lib/crt1.o`
 - `__cxx_global_var_init`
 - `_start()`
 - `main()`
 - `_end()`



- Application crashes

- Application crashes
- Generates a crash log

- Application crashes
- Generates a crash log

- Application crashes
- Generates a crash log

crashlog

Process: crash [28535]
User ID: 501

Date/Time: 2017-05-30 00:05:55.480 -0400
Crashed Thread: 0

Exception Type: EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: KERN_INVALID_ADDRESS at 0x000001234567
Termination Signal: Segmentation fault: 11

VM Regions Near 0x1234567:
__TEXT 0000000104eae000-0000000104eaf000 [4K] r-x/rwx

- Application crashes
- Generates a crash log

crashlog (2)

- Application crashes
- Generates a crash log

crashlog (2)

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread

0 crash 0x104eaefaf main + 4 (crash.cpp:13)
1 libdyld.dylib 0x7fffac8ef235 start + 1

Thread 0 crashed with X86 Thread State (64-bit):

rax: 0x0000000104eaefab	rbx: 0x0000000000000000
rcx: 0x00007fff5ad51c78	rdx: 0x00007fff5ad51bd0
rdi: 0x0000000000000001	rsi: 0x00007fff5ad51bc0
rbp: 0x00007fff5ad51ba0	rsp: 0x00007fff5ad51ba0
r8: 0x0000000000000000	r9: 0x0000000000b8baa5
r10: 0x00007ffb57e40d0	r11: 0xffffffff00000000
r12: 0x0000000000000000	r13: 0x0000000000000000
r14: 0x0000000000000000	r15: 0x0000000000000000
rip: 0x0000000104eaefaf	rfl: 0x0000000000010246
cr2: 0x00000000001234567	

crashlog (2)

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread

0 crash	0x104eaefaf main + 4 (crash.cpp:13)
1 libdyld.dylib	0x7fffac8ef235 start + 1

Thread 0 crashed with X86 Thread State (64-bit):

rax: 0x0000000104eaefab	rbx: 0x0000000000000000
rcx: 0x00007fff5ad51c78	rdx: 0x00007fff5ad51bd0
rdi: 0x0000000000000001	rsi: 0x00007fff5ad51bc0
rbp: 0x00007fff5ad51ba0	rsp: 0x00007fff5ad51ba0
r8: 0x0000000000000000	r9: 0x0000000000b8baa5
r10: 0x00007ffb57e40d0	r11: 0xffffffff00000000
r12: 0x0000000000000000	r13: 0x0000000000000000
r14: 0x0000000000000000	r15: 0x0000000000000000
rip: 0x0000000104eaefaf	rfl: 0x0000000000010246
cr2: 0x00000000001234567	

crashlog (3)

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread

0	crash	0x10e418faf	0x10e418000 + 4015
1	libdyld.dylib	0x7fffac8ef235	start + 1

Thread 0 crashed with X86 Thread State (64-bit):

rax: 0x000000010e418fab	rbx: 0x0000000000000000
rcx: 0x00007fff517e7c78	rdx: 0x00007fff517e7bd0
rdi: 0x0000000000000001	rsi: 0x00007fff517e7bc0
rbp: 0x00007fff517e7ba0	rsp: 0x00007fff517e7ba0
r8: 0x0000000000000000	r9: 0x0000000000b8bd80
r10: 0x00007ffb57e40d0	r11: 0xffffffff00000000
r12: 0x0000000000000000	r13: 0x0000000000000000
r14: 0x0000000000000000	r15: 0x0000000000000000
rip: 0x000000010e418faf	rfl: 0x0000000000010246
cr2: 0x0000000001234567	

crashlog (3)

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread

0 crash 0x10e418faf 0x10e418000 + 4015

1 libdyld.dylib 0x7fffac8ef235 start + 1

Thread 0 crashed with X86 Thread State (64-bit):

rax: 0x000000010e418fab	rbx: 0x0000000000000000
rcx: 0x00007fff517e7c78	rdx: 0x00007fff517e7bd0
rdi: 0x0000000000000001	rsi: 0x00007fff517e7bc0
rbp: 0x00007fff517e7ba0	rsp: 0x00007fff517e7ba0
r8: 0x0000000000000000	r9: 0x0000000000b8bd80
r10: 0x00007ffb57e40d0	r11: 0xffffffff00000000
r12: 0x0000000000000000	r13: 0x0000000000000000
r14: 0x0000000000000000	r15: 0x0000000000000000
rip: 0x000000010e418faf	rfl: 0x0000000000010246
cr2: 0x0000000001234567	

- Application crashes
- Generates a crash log

- Application crashes
- Generates a crash log

- Application crashes
- Generates a crash log

- Application crashes
- Generates a crash log
- Crash log contains addresses

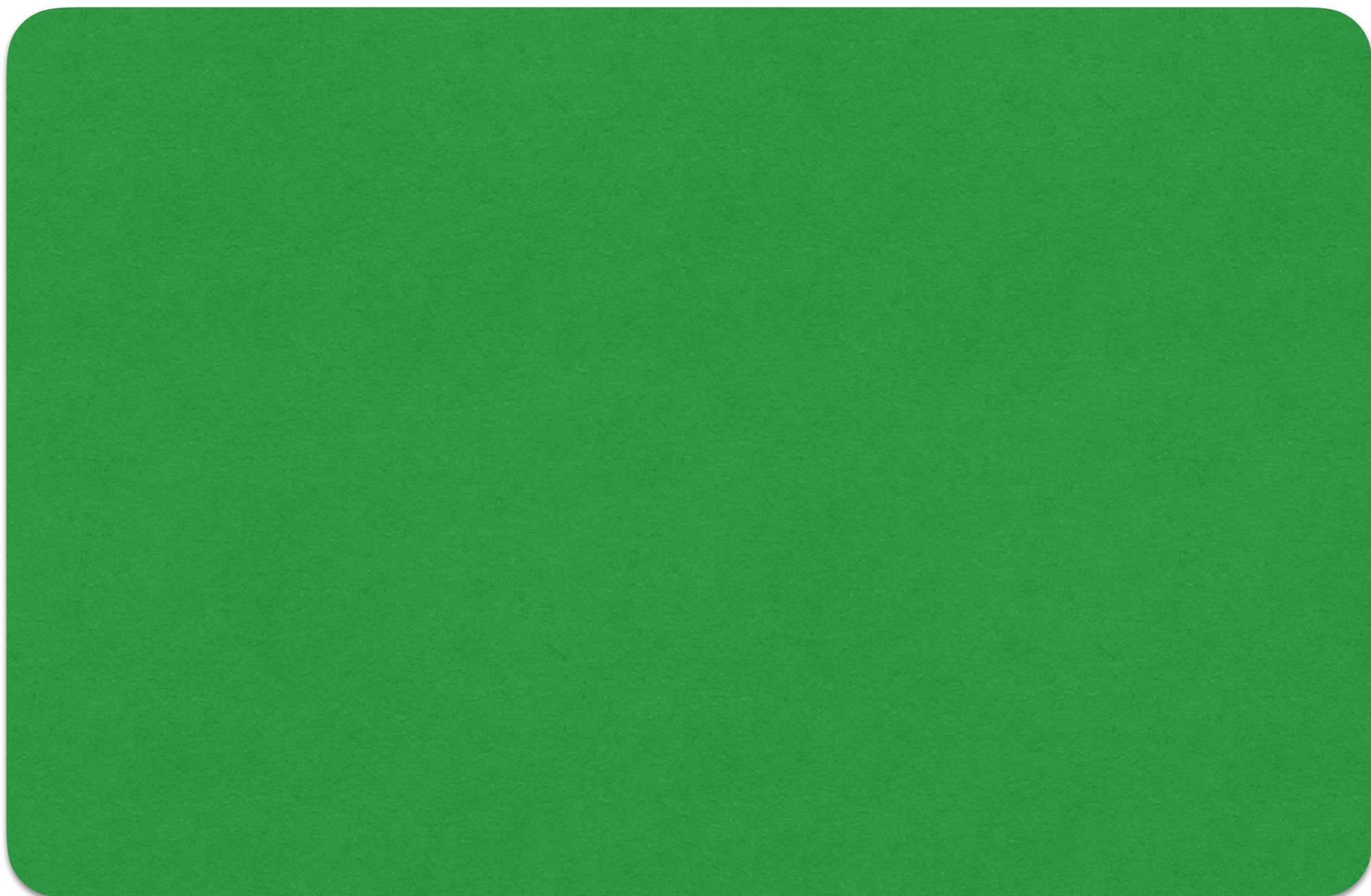
- Application crashes
- Generates a crash log
- Crash log contains addresses
- We must retrieve the symbols

- Application crashes
- Generates a crash log
- Crash log contains addresses
- We must retrieve the symbols

```
dsymutil crash -o crash.dSym; strip crash
```

- Application crashes
- Generates a crash log
- Crash log contains addresses
- We must retrieve the symbols

- Application crashes
- Generates a crash log
- Crash log contains addresses
- We must retrieve the symbols



- Symbolication

- Symbolication
 - atos

Symbolication

- Symbolication
 - atos

Symbolication

```
$ atos -o crash 0x00000010000faf  
main (in crash) (crash.cpp:13)
```

- Symbolication
 - atos

Symbolication

```
$ atos -o crash 0x00000010000faf  
main (in crash) (crash.cpp:13)
```

- Symbolication
 - atos

Symbolication

```
$ atos -o crash 0x00000010000faf  
main (in crash) (crash.cpp:13)
```

- Symbolication
 - atos

Symbolication

- Symbolication
 - atos

Symbolication

- Symbolication
 - atos

Symbolication

- Symbolication
 - atos
 - Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
```

```
(lldb) target create "crash"
```

```
Current executable set to 'crash' (x86_64).
```

```
(lldb) image list
```

```
[ 0] <UUID> 0x100000000 <PathToBinary>/crash  
    <PathToBinary>/crash.dSYM
```

```
[ 1] ...
```

```
(lldb) image lookup -a 0x100000faf
```

```
Address: crash[0x10000faf] (crash._TEXT._text + 23)
```

```
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13  
crash`main + 4 at crash.cpp:13
```

- Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash  
(lldb) target create "crash"  
Current executable set to 'crash' (x86_64).
```

```
(lldb) image list  
[ 0] <UUID> 0x100000000 <PathToBinary>/crash  
    <PathToBinary>/crash.dSYM  
[ 1] ...
```

```
(lldb) image lookup -a 0x100000faf  
Address: crash[0x10000faf] (crash._TEXT._text + 23)  
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13  
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash  
(lldb) target create "crash"  
Current executable set to 'crash' (x86_64).
```

```
(lldb) image list  
[ 0] <UUID> 0x100000000 <PathToBinary>/crash  
    <PathToBinary>/crash.dSYM  
[ 1] ...
```

```
(lldb) image lookup -a 0x100000faf  
Address: crash[0x10000faf] (crash._TEXT._text + 23)  
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13  
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash  
(lldb) target create "crash"  
Current executable set to 'crash' (x86_64).
```

```
(lldb) image list  
[ 0] <UUID> 0x100000000 <PathToBinary>/crash  
    <PathToBinary>/crash.dSYM  
[ 1] ...
```

```
(lldb) image lookup -a 0x10000FAF  
Address: crash[0x10000faf] (crash._TEXT._text + 23)  
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13  
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000FAF
Address: crash[0x100000tat] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

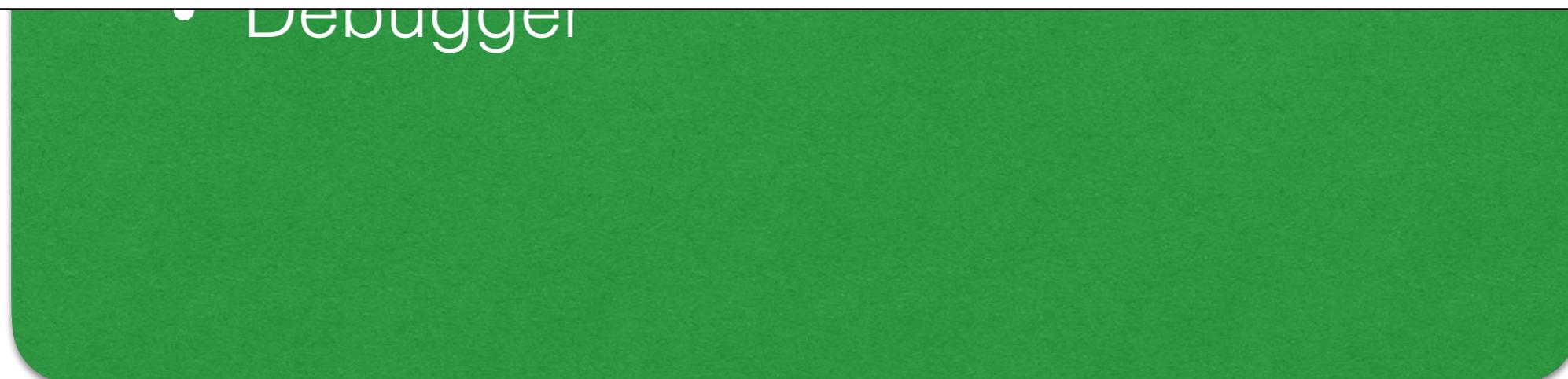
• Debugger

Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000FAF
Address: crash[0x10000fafl] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```



Symbolication

```
$ lldb crash
(lldb) target create "crash"
Current executable set to 'crash' (x86_64).

(lldb) image list
[ 0] <UUID> 0x100000000 <PathToBinary>/crash
    <PathToBinary>/crash.dSYM
[ 1] ...

(lldb) image lookup -a 0x100000faf
Address: crash[0x10000faf] (crash._TEXT._text + 23)
Summary: crash`main + 4 [inlined] dereference(int*) at crash.cpp:13
crash`main + 4 at crash.cpp:13
```

• Debugger

Symbolication

- Symbolication
 - atos
 - Debugger

ABI Calling Convention

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0 crash 0x10e418faf 0x10e418000 + 4015
1 libdyld.dylib 0x7fffac8ef235 start + 1

Thread 0 crashed with X86 Thread State (64-bit):
rax: 0x000000010e418fab rbx: 0x0000000000000000
rcx: 0x0007fff517e7c78 rdx: 0x0007fff517e7bd0
rdi: 0x0000000000000001 rsi: 0x0007fff517e7bc0
rbp: 0x0007fff517e7ba0 rsp: 0x0007fff517e7ba0
r8: 0x0000000000000000 r9: 0x000000000b8bd80
r10: 0x0007ffb57e40d0 r11: 0xfffffffff0000000
r12: 0x0000000000000000 r13: 0x0000000000000000
r14: 0x0000000000000000 r15: 0x0000000000000000
rip: 0x000000010e418faf rfl: 0x0000000000010246
cr2: 0x00000000001234567

ABI Calling Convention

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0 crash 0x10e418faf 0x10e418000 + 4015
1 libdyld.dylib 0x7fffac8ef235 start + 1

Thread 0 crashed with X86 Thread State (64-bit):
rax: 0x000000010e418fab rbx: 0x0000000000000000
rcx: 0x0007fff517e7c78 rdx: 0x0007fff517e7bd0
rdi: 0x0000000000000001 rsi: 0x0007fff517e7bc0
rbp: 0x0007fff517e7ba0 rsp: 0x0007fff517e7ba0
r8: 0x0000000000000000 r9: 0x000000000b8bd80
r10: 0x0007ffb57e40d0 r11: 0xfffffffff0000000
r12: 0x0000000000000000 r13: 0x0000000000000000
r14: 0x0000000000000000 r15: 0x0000000000000000
rip: 0x000000010e418faf rfl: 0x0000000000010246
cr2: 0x00000000001234567

ABI Calling Convention

Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
0 crash 0x10e418faf 0x10e418000 + 4015
1 libdyld.dylib 0x7fffac8ef235 start + 1

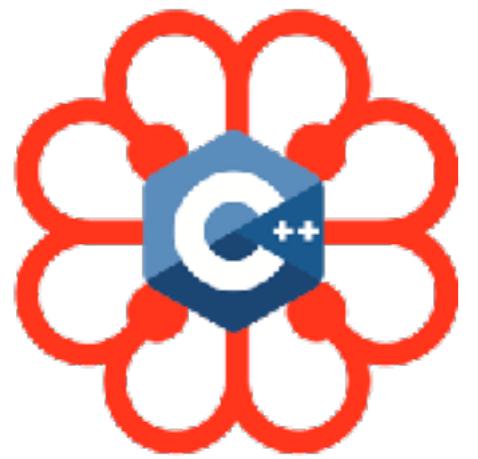
Thread 0 crashed with X86 Thread State (64-bit):
rax: 0x000000010e418fab rbx: 0x0000000000000000
rcx: 0x0007fff517e7c78 rdx: 0x0007fff517e7bd0
rdi: 0x0000000000000001 rsi: 0x0007fff517e7bc0
rbp: 0x0007fff517e7ba0 rsp: 0x0007fff517e7ba0
r8: 0x0000000000000000 r9: 0x000000000b8bd80
r10: 0x0007ffb57e40d0 r11: 0xfffffffff0000000
r12: 0x0000000000000000 r13: 0x0000000000000000
r14: 0x0000000000000000 r15: 0x0000000000000000
rip: 0x000000010e418faf rfl: 0x0000000000010246
cr2: 0x0000000001234567

ABI Calling Convention

- Symbolication
 - atos
 - Debugger
 - ABI Calling Convention

- Symbolication
 - atos
 - Debugger
 - ABI Calling Convention





Thank you