INDIRECT EXTERNAL ACCESS

H.J. Lu

Intel

August 2021

Issues with Copy Relocation

- On systems with copy relocation:
  - A copy in executable is created for the definition in a shared library at run-time by ld.so.
  - The copy is referenced by executable and shared libraries.
  - Executable can access the copy directly.
- Cons:
  - Overhead of a copy, time and space, may be visible at run-time.
  - Read-only data in the shared library becomes read-write copy in executable at run-time.
  - Local access to data with the STV_PROTECTED visibility in the shared library must use GOT.

Issues with Function Pointer

- On systems without function descriptor, function pointers vary depending on where and how the functions are defined.
  - If the function is defined in executable, it can be the address of function body.
  - If the function, including the function with STV_PROTECTED visibility, is defined in the shared library, it can be the address of the PLT entry in executable or shared library.
- Cons:
  - The address of function body may not be used as its function pointer.
  - ld.so needs to search loaded shared libraries for the function pointer of the function with STV_PROTECTED visibility.

Remove Copy Relocation

- Accesses, including in PIE and non-PIE, to undefined symbols must use GOT.
  - Linker may optimize out GOT access if the data is defined in PIE or non-PIE.
- Read-only data in the shared library remain read-only at run-time
- Address of global data with the STV_PROTECTED visibility in the shared library is the address of data body.
  - Can use IP-relative access.
  - Need GOT without IP-relative access.

Canonical Function Pointer

For systems without function descriptor:

- All global function pointers of undefined functions in PIE and non-PIE must use GOT.
  - Linker may optimize out GOT access if the function is defined in PIE or non-PIE.
- Function pointer of functions with the STV_PROTECTED visibility in executable and shared library is the address of function body.
  - Can use IP-relative access.
  - Need GOT without IP-relative access.
- Branches to undefined functions may use PLT.

Indirect External Access Marker

- Add GNU_PROPERTY_1_NEEDED
    - #define GNU_PROPERTY_1_NEEDED 0xb0008000
- Add GNU_PROPERTY_1_NEEDED_INDIRECT_EXTERN_ACCESS
    - #define GNU_PROPERTY_1_NEEDED_INDIRECT_EXTERN_ACCESS (1U << 0)
    - Protected symbol access within the shared library can be treated as local.
    - Copy relocation should be avoided at link-time and run-time.
    - GOT function pointer reference is required at link-time and run-time.

Compiler Support for Indirect External Access

- Add a compiler option, -fno-direct-extern-access:
  - Always to use GOT to access undefined symbols, including in PIE and non-PIE.
    - This is safe to do and doesn't break the ABI.
  - Generate an indirect external access marker in relocatable objects.
  - In executable and shared library, for symbols with the STV_PROTECTED visibility:
    - The address of data symbol is the address of data body.
    - For systems without function descriptor, the function pointer is the address of function body.
    - These break the ABI.
- -fdirect-extern-access, which is the default, disables this feature.

Linker Support for Indirect External Access

- If any relocatable input files contain the indirect external access marker:
  - Generate the indirect external access marker in output.
    - Linker should clear the indirect external access bit in executable when there are non-GOT or non-PLT relocations in relocatable input files without this bit set.
  - Avoid copy relocation if possible.
  - Access to symbols with the STV_PROTECTED visibility is the same as local access.
    - For systems without function descriptor:
      - Function pointer of functions is the address of function body.

Dynamic Linker for Indirect External Access

- Check the indirect external access marker on all components, the executable and its dependency shared libraries.
- Disallow copy relocation against definition with the STV_PROTECTED visibility in the shared library with the marker.
- For systems without function descriptor:
  - Disallow non-GOT function pointer reference in executable without the marker to the definition with the STV_PROTECTED visibility in a shared library with the marker.
  - Use the address of the function body as function pointer on functions with the STV_PROTECTED visibility, which are defined in shared libraries with the marker.

Software