# Why doesn't the MachO Patcher find any symbols in macOS 11 (Kernel Collections)?

### Beitrag von „hyux1“ vom 7. August 2024, 17:38

Someone is helping with the development, but there are problems. No symbols are found. I don't know what exactly he changed.

Can anyone say something about this?

I think it's this part of the code that causes it to fail. At least according to the log, but he didn't tell me.

Code

```
1. if (MachOInfo->Is64Bit) {
2.
3. struct nlist_64 *nlist64 = NULL;
4.
5. nlist64 = PTR_OFFSET(MachOInfo->LinkEditAddress, (symtabCommand->symoff - MachOInfo->LinkEditOffset) + MachOInfo->RelocBase, struct nlist_64 *);
6.
7. // Iterate the x86_64 Symbol List
8.
9. while (symbolIndex < symtabCommand->nsyms) {
10.
11. MachOUpdateSymbol (MachOInfo,
12. symbolString + nlist64->n_un.n_strx,
13. (UINT32)nlist64->n_value);
14.
15. symbolIndex++;
16. nlist64++;
17.
18. }
19.
20. } else {
21.
22. struct nlist *nlist = PTR_OFFSET(MachOInfo->LinkEditAddress, (symtabCommand->symoff - MachOInfo->LinkEditOffset) + MachOInfo->RelocBase, struct nlist *);
23.
```

```
24. // Iterate the i386 Symbol List
25.
26. while (symbolIndex < symtabCommand->nsyms) {
27.
28. MachOUpdateSymbol (MachOInfo,
29. symbolString + nlist->n_un.n_strx,
30. nlist->n_value);
31.
32. symbolIndex++;
33. nlist++;
34.
35. }
36.
37. }
```

Alles anzeigen

---

## Beitrag von „Tastenheld“ vom 8. August 2024, 11:27

The program correctly identifies the Mach-O file as 64-bit.

It struggles with parsing symbols, resulting in -1 symbols parsed and several null string issues.

The program finds entries in the kernelcache but fails to retrieve certain identifiers.

The patching process starts and completes, but an invalid parameter is encountered during the process.

These debug messages suggest issues with symbol parsing and parameter validation, which require further investigation and debugging to resolve.

Simplified explanation of the code:

The code checks if the Mach-O file is 64-bit or 32-bit.

It calculates the starting address of the symbol table.

It iterates through each symbol in the table, updating their information using a helper function.

Different structures are used for 64-bit (nlist_64) and 32-bit (nlist) architectures.

However - you missed to tell us one important thing... Which Hardware do you use?

---

### Beitrag von „pupokass“ vom 8. August 2024, 11:35

I tested it on GA-Z77-DS3H rev 1.1 i5 3570k intel hd4000