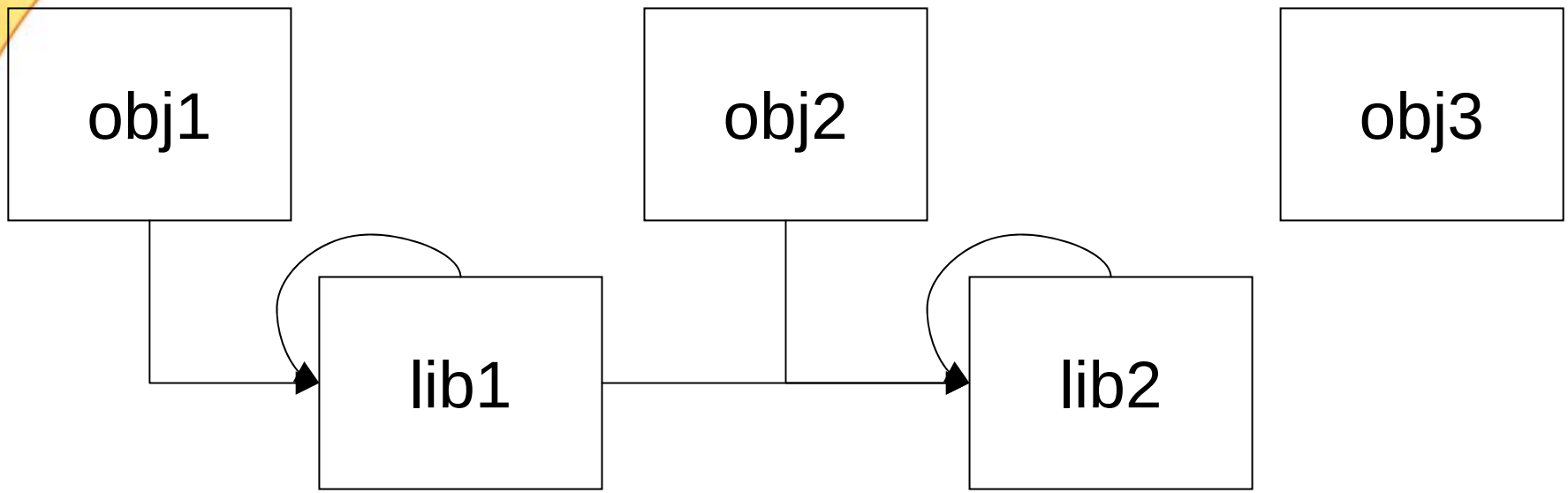
Linking Libraries

- Unresolved references kept in running list
- Each file seen adds/subtracts from that list
- When library is opened, include modules that resolve references in the current list
- Module from a library can also introduce references that must be resolved

Default Library Search



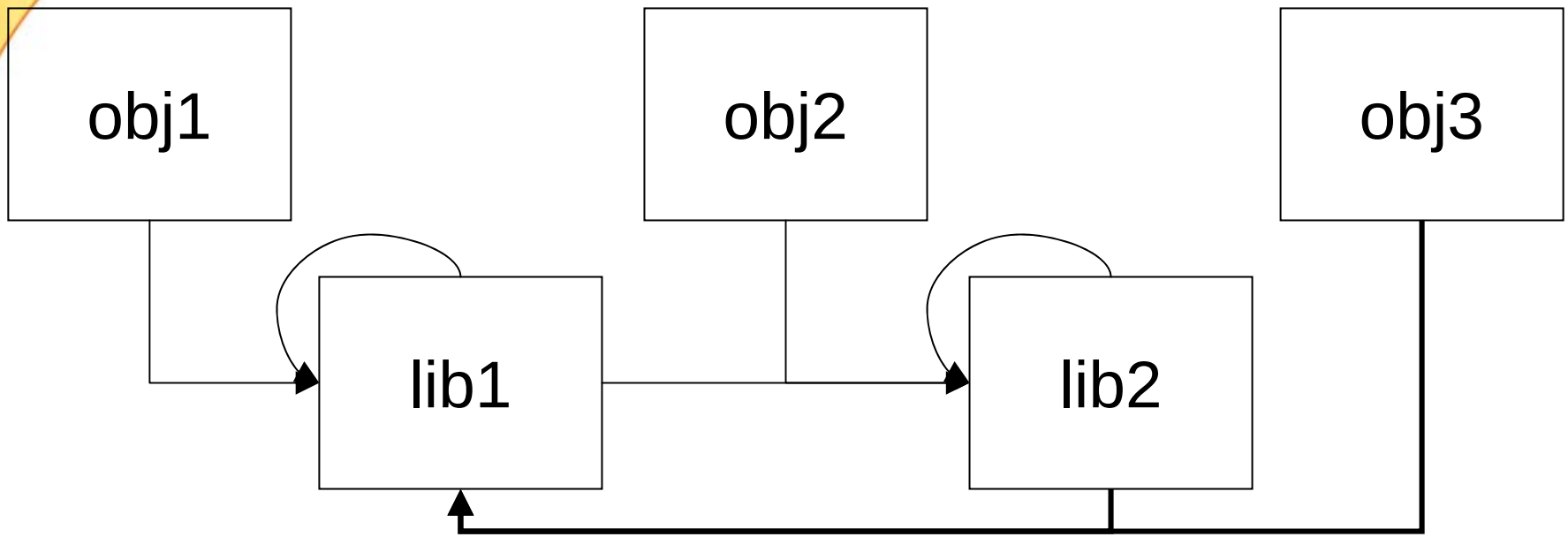

```
lnkxx obj1 lib1 obj2 lib2 obj3
```

- Arrows represent series of lib(s) searched

Default Library Search

- All references between obj* files will resolve, regardless of ref/def order
- Obj1 will resolve refs from both lib1 and lib2
- Lib1 will resolve refs from lib1 and lib2
- Obj2 will resolve refs only from lib2, not from lib1
- Lib2 will resolve refs only from lib2, not lib1
- Obj3 will not resolve any references from lib1 or lib2

Exhaustive Search -x



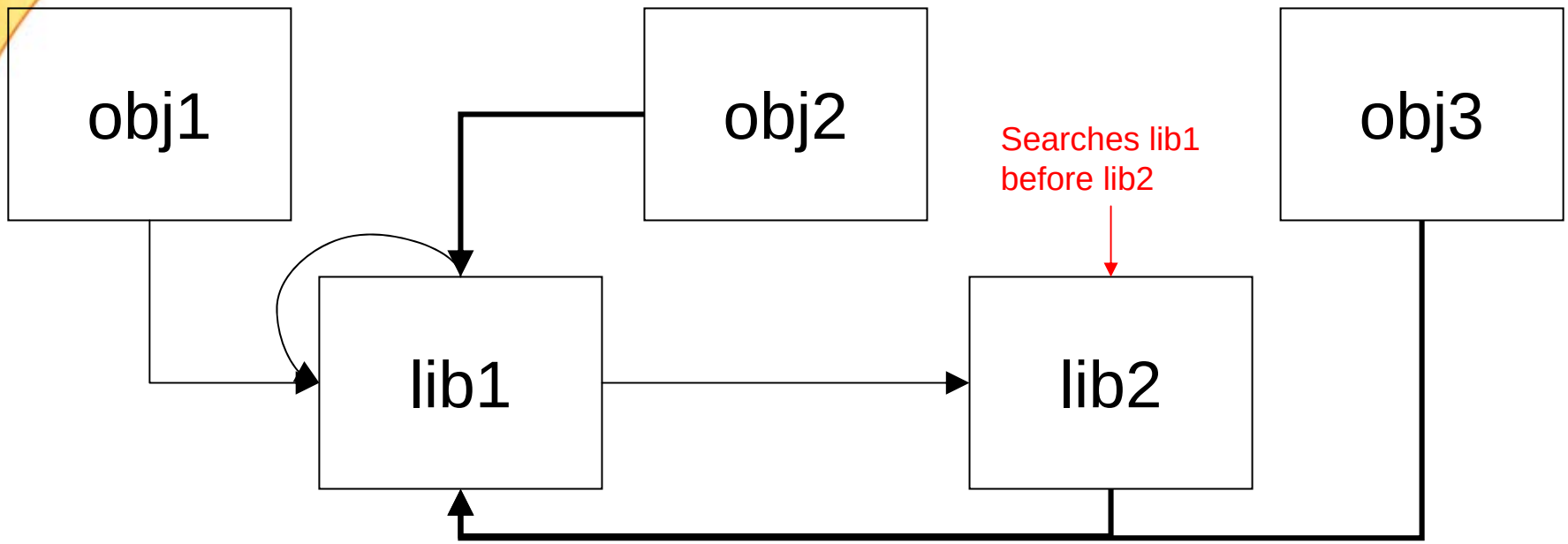

```
lnkxx -x obj1 lib1 obj2 lib2 obj3
```

- Bold arrows denote difference vs. default

Exhaustive Search -x

- All obj* files will resolve refs in both libraries. Only the order is different.
- Obj1: lib1 then lib2
- Lib1: lib1 then lib2
- Obj2: lib2 then lib1
- Lib2: lib2 then lib1
- Obj3: lib1 then lib2

Priority Search `-priority`




```
lnkxx -priority obj1 lib1 obj2 lib2 obj3
```

- Bold arrows denote difference vs. default

Priority Search `-priority`

- Everyone resolves library references from lib1, then lib2
- Especially note that lib2 resolves refs from lib1 before searching itself
- Poster child example: BIOS libs redefining the memory allocation routines malloc/free. Those calls always need to be resolved from the BIOS lib, and then the compiler RTS lib.