

Erstellen von Bibliotheken

Was ist eine Library?

Ein Paket von kompiliertem Code und Daten, das später in ein Programm eingefügt werden soll.

Wofür?

- modularere Programmgestaltung
- leichter updatebar

Arten:

- static - statische Bibliotheken - benötigte Bibliotheksinhalte werden vom Linker in die ausführbare Programmdatei eingefügt
- shared - gemeinsam genutzte Bibliotheken - werden vor der Ausführung geladen.
- (dynamically loaded - werden dynamisch bei der Ausführung geladen, lädt "normale" static oder shared libraries.)

Static Libraries

- Zusammenfassung/Archiv von Objektfiles
- enden auf ".a"
- enthält kompilierten Code => wiederholtes Kompilieren desselben Quellcodes kann vermieden werden
- sinnvoll, wenn die Library, aber nicht ihr Quelltext veröffentlicht werden soll.

Erstellen:

- Objectfile erstellen mit Flag `-c` (nicht linkern): `cc -c FILE.c`
- Objektdaten packen: `ar rcs libNAME.a FILE1.o [FILE2.o] *`

Bedeutungen der `ar` Befehle sind auf der Folie 5 aufgelistet.

Verwenden:

```
gcc .... -LDIR -lNAME
```

(DIR: Verzeichnis der Bibliothek; NAME: Name der Bibliothek)

Shared Libraries

- werden beim Programmstart geladen
- werden bei mehreren gleichzeitig nutzenden Programm nur EINMAL in den Arbeitsspeicher geladen
- Bibliothek einfacher zu updaten, eigene verwendende Programme leichter zu warten
- Updates: ausführbare Programmdatei muss nicht neu erzeugt werden
- Programmdatei ist kleiner

Namensgebung: (Bibliothek NAME, Version X, Minor Version Y, Release Z)

- "soname" `libNAME.so.X`
 ("fully qualified soname" `/path/to/lib/libNAME.so.X`)
- "real name" `libNAME.so.X.Y[.Z]`
- "linker name" `libNAME.so` - wird vom Linker gesucht.

Alle drei sollten im Dateisystem vorhanden sein – dabei sind soname und linker name Links auf den real name.

Normale, systemweite Installation in `/usr/lib` und `/usr/local/lib`

Shared Libraries

Linken gegen Bibliothek mit `-lNAME`.

Liegt Bibliothek in einem anderen Verzeichnis, Suchpfad beim Linken mitangeben:

`-LDIR`

Erstellen:

- Objectfile erstellen mit `-fPIC`: `gcc -fPIC -c NAME.c`
- Bibliothek erstellen: `gcc -shared -Wl,-soname,SONAME -o REALNAME FILELIST LIBLIST`
- (Installieren: nach `/usr/lib` oder `/usr/local/lib` kopieren, `ldconfig -n DIR`)
- Verwenden von nicht installierten Bibliotheken (ALTERNATIVEN):
 - I.
 1. Programm kompilieren mit `-LDIR`
 2. Speichern von `DIR` in `LD_LIBRARY_PATH` oder Aufrufen mit `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:DIR`
 - II. Kompilieren mit Linkeroption `--rpath`: `gcc -Wl,--rpath,DIR ...`

Hilfsmittel

ldd zeigt die shared libraries die vom angegebenen Programm geladen werden und deren Ort im Dateisystem an.

nm <lib|prg|obj>* gibt die Liste der in der angegebenen kompilierten Datei(en) verwendeten Symbole aus.

ar [-]befehl[mod] archive [member...]

Wichtige Befehle:

- t – listet die im archive gepackte Bibliotheken auf.
- d – löscht angegebene member aus archive.
- r – fügt member an und ersetzt dabei bereits vorhandene, gleichnamige member.

Wichtige Modifiers (können beliebig kombiniert werden):

- s – legt einen Dateiindex an oder aktualisiert diesen.
- v – "geschwätzig", gibt zusätzliche Informationen aus.
- c – schaltet die Warnings beim Anlegen neuer Archive aus.