

Programmeertechnieken 2017

Huiswerk 1: Building and linking a library

Deadline: Vrijdag 17 februari, voor het einde van de dag.

1 Omschrijving

Het doel van deze huiswerkopgave is om zelf een shared library te maken en een testprogramma te schrijven dat met deze library wordt “gelinkt”. De shared library moet ten minste *drie* functies bevatten. Zorg ervoor dat de functies verschillende argumenten en returnwaarden hebben. De functies moeten uiteraard zijn gedeclareerd in een aparte header-file. Schrijf een *Makefile* om de library te compileren.

Vervolgens schrijf je een testprogramma dat alle functies in de library aanroept. In het testprogramma doe je een `#include` van de header-file van de library. Daarnaast moet het testprogramma worden gelinkt met de library zodat de door het testprogramma aangeroepen functies kunnen worden gevonden. Ook voor het testprogramma moet een nette *Makefile* worden geschreven.

Als houvast kun je de volgende directory-structuur als voorbeeld gebruiken. De bestanden gemarkeerd met (*) zijn een resultaat van het uitvoeren van de *Makefile* en hoeven niet te worden ingeleverd:

```
hw1/lib/include/mylib.h
hw1/lib/src/mylib.c
hw1/lib/src/mylib.o (*)
hw1/lib/Makefile
hw1/lib/libmine.so (*)
hw1/app/test.c
hw1/app/Makefile
hw1/app/test (*)
```

2 Inleveren

Er mag worden gewerkt in tweetallen. Zorg ervoor dat alle bestanden die worden ingeleverd (source code, *Makefiles*) zijn voorzien van naam en studentnummer! Object files, shared objects en executables (kortom: alles dat wordt gegenereerd door een *Makefile*) hoeven *niet* te worden ingeleverd. Maak een “gzipped tar” bestand van je directory-structuur:

```
tar -czvf hw1-sXXXXXXX-sYYYYYYY.tar.gz hw1/
```

Vul op de plek van `XXXXXXX` en `YYYYYYY` de bijbehorende studentnummers in. De inzendingen kunnen worden verzonden per e-mail naar `pt2017 (at) handin.liacs.nl` met als onderwerp “PT Huiswerk 1”.

3 Aanwijzingen en Tips

- Gebruik bij het compileren van de object-bestanden de `-I` parameter om het pad naar de header-bestanden op te geven.
- Vergeet niet `-fPIC` als optie te gebruiken bij het compileren van de object-bestanden die onderdeel zullen worden van een shared library.
- Het maken van een shared library doe je met de `-shared` optie van `gcc`. Bijvoorbeeld:

```
gcc -shared -fPIC -o libmine.so src/obj1.o src/obj2.o src/objn.o
```

- Maak gebruik van variabelen in *Makefiles* en ook de speciale variabelen als `$(C)`, `$(^)`, `$(<`.

- Zorg voor een *clean* target in de Makefiles.
- Bij het linken van het testprogramma moet *gcc* weten in elke directory *gcc* jouw eigen library kan vinden. Dit kun je doen met de optie *-L*. Daarnaast geef je met *-l* op welke library moet worden gelinkt. *gcc* zal zelf *lib* voor de naam zetten bij het zoeken. Bijvoorbeeld:

```
gcc -Wall -L../lib -o test test.c -lmine
```

- Met het programma *ldd* kun je nagaan wat voor dependencies een programma of shared library heeft. Je kunt bijvoorbeeld voor je testprogramma nagaan of deze inderdaad een dependency heeft op *libmine.so* en of *libmine* wel wordt gevonden.
- Zeer waarschijnlijk zal *gcc* zonder problemen het testprogramma compileren, maar kun je het daarna niet uitvoeren. Je krijgt bijvoorbeeld de foutmelding “cannot open shared object file: No such file or directory”. Dit kan als volgt worden verklaard: bij het opstarten van je programma zal de dynamische linker gaan zoeken naar *libmine.so*. Het zal daarbij kijken op een aantal standaard locaties zoals */lib* en */usr/lib*. Wij hebben onze library echter niet op een van deze standaard locaties geïnstalleerd en daarom moeten we aan de dynamic linker bekend maken waar deze moet zoeken. Dit doen we met de *LD_LIBRARY_PATH* environment variabele. We kunnen ook met *ldd* nagaan of onze library gevonden kan worden. Bijvoorbeeld:

```
$ export LD_LIBRARY_PATH="../lib"
$ ldd ./test
linux-vdso.so.1 => (0x00007ffed3d2f000)
libmine.so => ../lib/libmine.so (0x00007fd986a83000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd9866a5000)
/lib64/ld-linux-x86-64.so.2 (0x00007fd986c87000)
```