SO Cheat Sheet
Lab 1 - Introducere

Linux

gcc

```
gcc [-c|-S|-E] [-std=standard]
    [-g] [-pg] [-Olevel]
    [-Wwarn...] [-pedantic]
    [-Idir...] [-Ldir...]
    [-Dmacro[=defn]...] [-Umacro]
    [-o outfile] [@file] infile...
```

Fazele compilării

- **-E**, preprocesare fişier sursă
  - gcc -o hello.i -E hello.c
- **-S**, compilare fişier sursă
  - gcc -o hello.s -S hello.c
- **-c**, asamblare fişier sursă
  - gcc -o hello.o -c hello.c

Alte opţiuni

- **-Wall**, activarea tuturor avertismentelor.
- **-lxlib**, caută biblioteca xlib la editarea de legaturi.
- **-Ldir**, adaugă dir la lista directoarelor căutate pentru -l
- **-Idir**, adaugă dir la începutul listei de căutare pentru fişierele antet.
- **-DMACRO[=val]**, defineşte macro în linia de comanda.
- **-g**, genereare simboluri de debug folosite mai târziu de debugger.

make

```
make [ -f makefile ] [ options ] ... [ targets ]
```

```
target: dependinte
<tab>    comanda
```

Variabile uzuale

- **CC**, defineşte compilatorul folosit
- **CFLAGS**, defineşte flag-urile de compilare
- **LDLIBS**, defineşte bibliotecile folosite la editarea de legături
- **$@**, se expandează la numele target-ului
- **$^**, se expandează la lista de dependinţe
- **$<**, se expandează la prima dependinţă

gdb

```
gdb prog [arguments]
```

Executabilul trebuie compilat cu flag-ul **-g**

Comenzi:

- **b [file:]function**, setează brakepoint la funcţia dată.
- **r arglist**, run - rulează programul cu argumentele date.
- **bt**, backtrace - afişează stiva programului
- **p expr**, print - afişează valoarea lui expr
- **c**, continue - continuă rularea programului după oprirea la un breakpoint
- **n**, next - execută următoare linie de program, trece peste orice apel de funcţie pe acea linie
- **s**, step - execută următoarea linie de program, trece prin orice apel de funcţie de pe acea linie.
- **quit**, ieşire din GDB.

Creare biblioteci

- **statice**
  - creare arhivă
    - **ar rc libX_static.a X1.o X2.o**
  - legare bibliotecă
    - **gcc -Wall main.c -o main -lX_static -L.**
- **dinamice**
  - **creare obiect partajat**
    - **gcc -shared f1.o f2.o -o libX_shared.so**
  - **legare bibliotecă**
    - `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.`
    - **gcc -Wall main.c -o main -lX_shared -L.**

Windows

cl

```
CL [option...] file... [option | file]... [lib...]  [/link link-opt
```

Opţiuni

- **/c**, realizează doar compilare fără editare de legături.
- **/Wall**, activează toate avertismentele
- **/D**, definire macrou în linie de comandă
- **/I<dir>**, adaugă dir la începutul listei de căutare pentru fişierele antet.
- **/LIBPATH<dir>**, indică editorului de legături să caute şi în dir bibliotecile pe care le va folosi programul

Specificare fişiere de ieşire

- **/Fa<file>**, specifică numele fişierului în limbaj de asamblare.
- **/Fo<file>**, specifică numele fişierului obiect.
- **/Fe<file>**, specifică numele fişierului executabil.

nmake

```
NMAKE [option...] [macros...] [targets...]
```

- **/F**, pentru a rula alt fişier make decât cel implicit cu numele Makefile.

Creare biblioteci

- **statice**
  - se foloseşte comanda **lib**
  - **lib /out:intro.lib f1.obj f2.obj**
- **dinamice**
  - precizarea explicită a simbolurilor folosite
    - **__declspec(dllimport)**, folosit pentru a importa o funcţie
    - **__declspec(dllexport)**, folosit pentru a exporta o funcţie
  - creare bibliotecă dinamică şi bibliotecă de import
    - folosind opţiunea **/LD** a compilatorului cl
      - **cl /LD f1.obj f2.obj**
    - folosind comanda link
      - **link /nologo /dll /out:intro.dll /implib:intro.lib f1.obj f2.obj**