

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

Alte opțiuni

- **-Wall**, activarea tuturor avertismentelor.
- **-l_xlib**, 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 fișierele antet.
- **-DMACRO[=val]**, definește macro în linia de comanda.
- **-g**, generare simboluri de debug folosite mai târziu de debugger.

make

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

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

Variabile uzuale

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

gdb

```
gdb prog [arguments]
```

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

Comenzi:

- **b [file:]function**, setează breakpoint la funcția dată.
- **r arglist**, run - rulează programul cu argumentele date.
- **bt**, backtrace - afișează stiva programului
- **p expr**, print - afișează 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 funcție pe acea linie
- **s**, step - execută următoarea linie de program, trece prin orice apel de funcție de pe acea linie.
- **quit**, ieșire 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]
```

Opțiuni

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

Specificare fișiere de ieșire

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

nmake

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

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

Creare biblioteci

- **statice**

– se folosește comanda **lib**

– **lib /out:intro.lib f1.obj f2.obj**

- **dinamice**

– precizarea explicită a simbolurilor folosite

```
* __declspec(dllimport), folosit pentru a importa o funcție
```

```
* __declspec(dllexport), folosit pentru a exporta o funcție
```

– creare bibliotecă dinamică și bibliotecă de import

```
* folosind opțiunea /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
```