Laborator 1

Introducere

Sisteme de Operare

17-23 Februarie 2011

De unde pornim?

- Welcome to SO!
- http://elf.cs.pub.ro/so

Notare

- curs
  - 4 lucrări X 0.5 puncte, 2 puncte
  - final, 3 puncte
- teme de casă
  - $1 + 5 + 5 = 11$ puncte + corelare curs
- laborator
  - nu are pondere în nota finală
  - obligatoriu prezență activă la cel puțin 8 laboratoare
- http://elf.cs.pub.ro/so/wiki/reguli-notare

Desfășurare laborator

- parcurgere laborator acasă, 40 de minute
- prezentare teoretică + întrebări, 20 de minute
- rezolvare exerciții, 100 de minute
- punctaj între 0 și 11
- bucuria rezolvării unui laborator de SO infinită

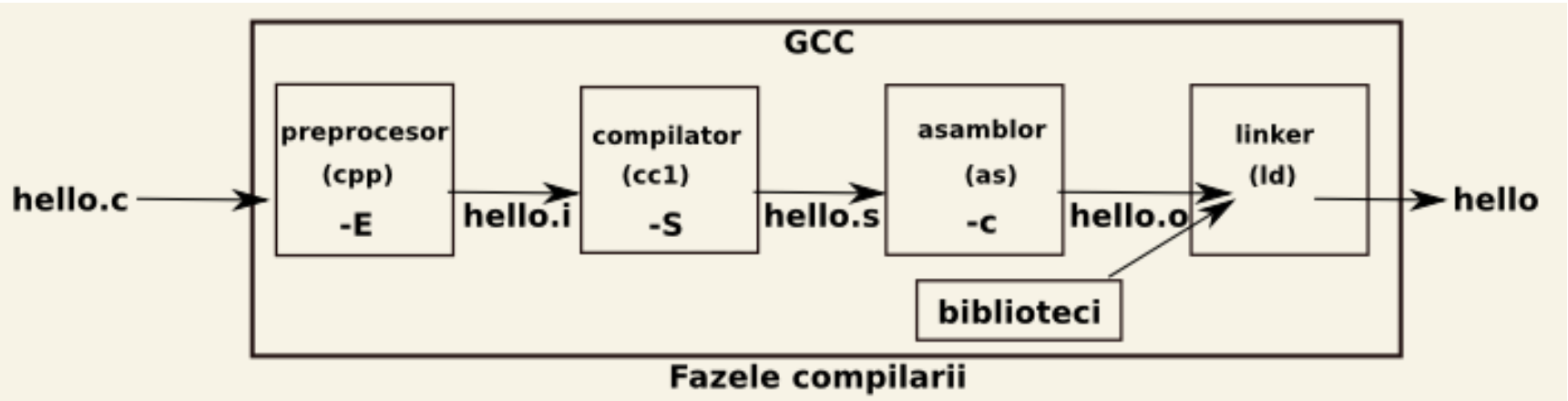
Suport de laborator

- cărți
  - TLPI, The Linux Programming Interface, M. Kerrisk
  - WSP4, Windows System Programming 4th Edition, J. Hart
- listă de discuții
  - http://cursuri.cs.pub.ro/cgi-bin/mailmain/listinfo/so
- canal IRC, rețea Freenode, #cs_so

Ce vom învăța în laborator

- Compilare, depanare, biblioteci
- Operații I/E simple
- Procese
- Gestiunea memoriei
- Comunicarea inter-procese
- Semnale
- Memoria virtuală
- Fire de execuție (2)
- Operații de I/E avansate (2)
- Profiling
- Securitate

- compilare
  - traducerea unui program (limbaj sursă, limbaj țintă)
- makefile
  - automatizarea procesului de compilare
- depanare
  - detectarea erorilor din programe
- biblioteci
  - colecție de fișiere precompilate


GCC
hello.c
preprocesor
(cpp)
-E
hello.i
compilator
(cc1)
-S
hello.s
asamblor
(as)
-c
hello.o
linker
(ld)
hello
biblioteci
Fazele compilarii

GCC

- GNU Compiler Collection
- gcc hello.c
  - compilare simplă, rezultă fișierul executabil a.out
- gcc hello.c -o hello
  - compilare simplă cu specificarea numelui fișierului de ieșire
- gcc hello.c -c -o hello.o
  - oprirea compilării după obținerea fișierului obiect
- gcc hello.o -o hello
  - editarea de legături pentru fișierul obiect hello.o

cl

- cl.exe - Microsoft Compiler
- cl hello.c
  - compilare simplă, rezultă fișierul executabil hello.exe
- cl /Fehello_win.exe hello.c
  - compilare simplă cu specificarea numelui executabilului
- cl /c hello.c
  - obținerea fișierului obiect
- cl /Fehello.obj
  - editarea de legături pentru fișierul obiect
- cl /? - help

Fișiere make

- automatizarea compilării
- fișier Makefile
  - reguli
  - comenzi
  - variabile
- compilare 'deșteaptă'
- make vs nmake

gdb

- fișierele sunt compilate cu opțiunea -g
- execuție
  - gdb ./a.out
- comenzi utile
  - p - print
  - bt - backtrace
  - step, next
  - set args

Biblioteci

- statice
  - rezolvare simboluri în momentul editării de legături
  - funcțiile utilizate sunt incluse în executabil
  - dimensiune executabil mai mare, rulare mai rapidă
- dinamice
  - rezolvare simbolurilor se poate face
    - la încărcare (load-time)
    - la rulare (run-time) (dlopen and friends)
  - executabil de dimensiune redusă

Lucrul cu biblioteci în Linux

- crearea unei biblioteci statice (.a)
  - ar rc libxyz.a f1.o f2.o
- crearea unei biblioteci partajate (.so)
  - gcc -fPIC -c f1.c
  - gcc -shared f1.o -o libxyz.so
- legarea cu o bibliotecă
  - -lxyz
  - -Lpath
  - LD_LIBRARY_PATH

Lucrul cu biblioteci în Windows

- crearea unei biblioteci statice (.lib)
  - lib /out:<nume.lib> <lista fisiere obiect>
- crearea unei biblioteci dinamice (.dll)
  - __declspec(dllimport), __declspec(dllexport)
  - link (/dll) sau cl /LD

Întrebări

- Care este cea mai simplă comandă care compilează programul hello.c?
- De ce includerea de mai multe ori a unui fișier antet standard nu generează o eroare la compilare?
- Este posibil ca dimensiunea unui fișier după faza de preprocesare să fie mai mică decât a fișierului .c original?