Laborator 1

Introducere

Sisteme de Operare

25 Februarie - 2 Martie 2016

Cine sunt?

- email
- experiență
- pasiuni relevante
- de ce SO?

Resurse

- Wiki: http://ocw.cs.pub.ro/so
  - NeedToKnow page: http://ocw.cs.pub.ro/so/meta/need-to-know
  - Folosiți feed-ul RSS
- Lista de discuții
  - so@cursuri.cs.pub.ro
  - Abonați-vă (detalii pe wiki)
- Catalog Google, calendar Google
- Mașini virtuale
- vmchecker (verificare teme)
- Documentație
- cs.curs.pub.ro (rol de portal + workshop)
- Pagină de Facebook

Despre laborator

- Subiecte principale
  - Procese
  - Thread-uri
  - Comunicare și sincronizare
  - Memorie
  - Sisteme de fișiere
  - I/O
- POSIX/Win32 API programming (C/C++)
- 7 minute workshop / 15 min prezentare / restul task-uri
- Tutorial-like, task-based, learn by doing
- Karma Points ("pentru cei puternici")

Workshop

- Testul
  - 3 întrebări din laboratorul curent
  - Primele 7 minute din laborator
  - Întrebări atât teoretice, cât și practice
- Punctare
  - Corectați voi: acasă, random și anonim câte două teste; deadline: o săptămână după încheierea laboratorului
  - Nota finală pe test: punctajul primit pe test (50%) + punctaj pe cum ați corectat (50%)
- Total teste: 5 (la începutul laboratoarelor 2, 4, 6, 8, 11)

Teme

- Tema 0 – hash-table
- Tema 1 – mini-shell
- Tema 2 – demand pager/swapper
- Tema 3 – thread scheduler
- Tema 4 – server de fișiere

- Intense
- Necesare: aprofundare API (laborator) și concepte (curs)
- Estimare de timp: 8-20 ore pe temă
- Teste publice
- Suport de testare la submit - feedback imediat

Reguli și notare

- http://ocw.cs.pub.ro/courses/so/meta/notare/reguli-notare-cb
- Examen final - 4 puncte
- Activitate laborator - 1 punct
  - 0.75 puncte workshop, 0.75 puncte task-uri => trunchiere la 1 punct
  - Prezență activă obligatorie la cel puțin 8 laboratoare pentru a intra în examen

Reguli și notare (2)

- Teme - 5 puncte + 4 puncte * corelare punctaj
  - Fiecare temă valorează 1punct
  - Rezolvată pe ambele platforme, fiecare temă este punctată cu maxim 1 punct, punctajele se cumulează și se trunchiază la 1 punct.
  - Trunchiere la 5 puncte pentru teme
- Depunctare teme
  - -0.25 puncte pe zi (din 10) timp de 14 zile
  - După 14 zile tema nu se mai punctează
- Punctajul de absolvire a cursului este 4.5
- După restanțe tot punctajul se resetează la 0

Karma Awards

- Cum se obțin Karma Points?
  - Participare la discuțiile din timpul cursului
  - Participare la discuțiile din timpul laboratorului
  - Răspunsuri pe lista de discuții
  - Editarea wiki-ului
  - Exercițiile bonus din timpul laboratorului
  - Teme elegante
    - Coding style consistent, comentarii punctuale, claritatea codului
    - Soluții simple și corecte
    - Modularitate, cursivitate

Desfășurare laborator

- Parcurgere laborator acasă - 40 de minute
- Workshop - 7 minute
- Prezentare teoretică + întrebări - 15 de minute
- Rezolvare exerciții - 80 de minute
  - Punctaj între 0 și 11
  - Bucuria rezolvării unui laborator de SO infinită :)
- Workshop
  - 3 întrebări din laboratorul curent

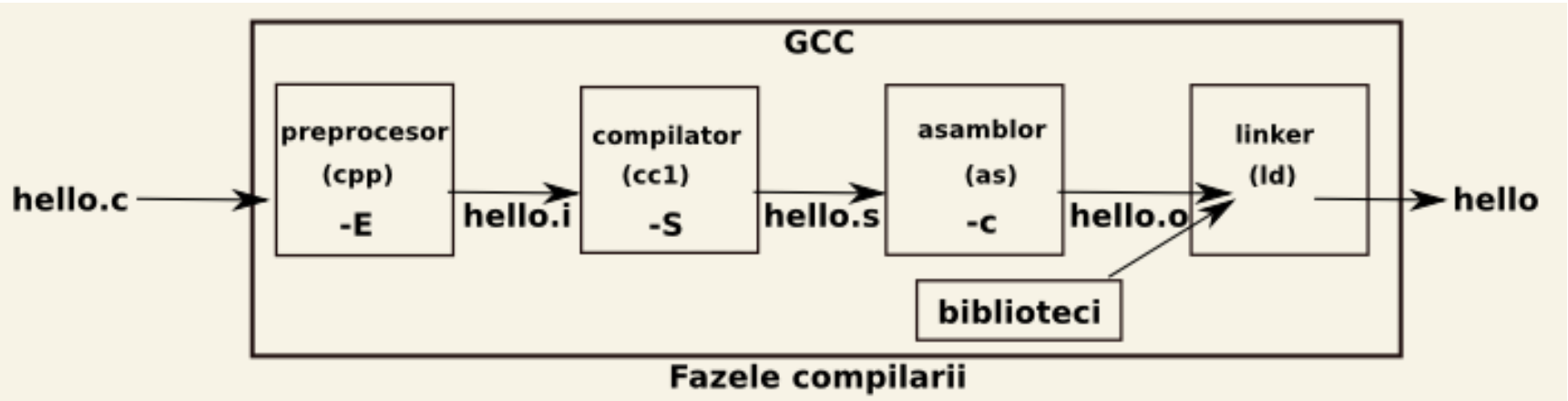
- Cărți
  - TLPI, The Linux Programming Interface, M. Kerrisk
  - WSP4, Windows System Programming 4th Edition, J. Hart
- Listă de discuții
  - http://cursuri.cs.pub.ro/cgi-bin/mailman/listinfo/so
- Canal IRC, rețea Freenode, #cs_so

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

Laboratorul 1

- 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