

Laborator 1

Introducere

Sisteme de Operare

23 Februarie - 1 Martie 2017

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

- ▶ 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)
- ▶ Pagină de Facebook

- ▶ Subiecte principale
 - ▶ Procese
 - ▶ Thread-uri
 - ▶ Comunicare și sincronizare
 - ▶ Memorie
 - ▶ Sisteme de fișiere
 - ▶ I/O

- ▶ POSIX/Win32 API programming (C/C++)
- ▶ prezentare + joc interactiv
- ▶ Tutorial-like, task-based, learn by doing
- ▶ Karma Points ("pentru cei puternici")

- ▶ Tema 1 – hash-table
- ▶ Tema 2 – mini-shell
- ▶ Tema 3 – demand pager/swapper
- ▶ Tema 4 – thread scheduler
- ▶ Tema 5 – 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

- ▶ <https://ocw.cs.pub.ro/courses/so/meta/notare/reguli-notare-ca-cb-cc>

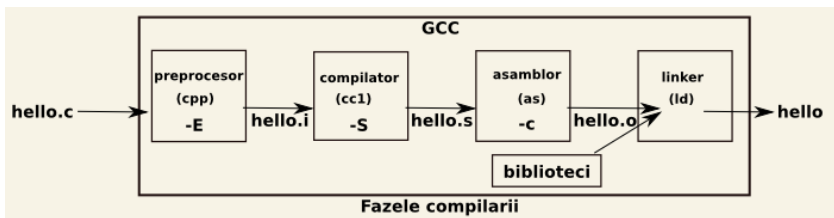
- ▶ Cum se obțin Karma Points?
 - ▶ Participare la discuțiile din timpul cursului
 - ▶ Participare la discuțiile din timpul laboratorului
 - ▶ Răspunsuri pe lista de discuții
 - ▶ Editarea wiki-ului
 - ▶ Exercițiile bonus din timpul laboratorului
 - ▶ Teme elegante
 - ▶ Coding style consistent, comentarii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

- ▶ Parcurgere laborator acasă - 40 de minute
- ▶ Prezentare teoretică + joc introductiv - 30 de minute
- ▶ Rezolvare exerciții - 80 de minute
 - ▶ Punctaj între 0 și 11
 - ▶ Bucuria rezolvării unui laborator de SO infinită :)

- ▶ 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

- ▶ Compilare, depanare, biblioteci
- ▶ Operații I/E simple
- ▶ Procese
- ▶ Semnale
- ▶ Gestiunea memoriei
- ▶ Debugging
- ▶ Memoria virtuală
- ▶ Fire de execuție (2)
- ▶ Operații de I/E avansate (2)
- ▶ Sisteme de fișiere

- ▶ Compilare
 - ▶ Traducerea unui program (limbaj sursă, limbaj țintă)
- ▶ Makefile
 - ▶ Automatizarea procesului de compilare
- ▶ Depanare
 - ▶ Detectarea erorilor din programe
- ▶ Biblioteci
 - ▶ Colecție de fișiere precompilate



- ▶ 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.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.exe hello.obj
 - ▶ Editarea de legături pentru fișierul obiect
- ▶ cl /? - help

- ▶ Automatizarea compilării
- ▶ Fișier Makefile
 - ▶ Reguli
 - ▶ Comenzi
 - ▶ Variabile
- ▶ Compilare 'deșteaptă'
- ▶ make vs. nmake

- ▶ Fişierele sunt compilate cu opţiunea -g
- ▶ Execuţie
 - ▶ `gdb ./a.out`
- ▶ Comenzi utile
 - ▶ `p` - print
 - ▶ `bt` - backtrace
 - ▶ `step`, `next`
 - ▶ `set args`

- ▶ 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

- ▶ 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`

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