PB173 Linux

01 GCC, Make

Roman Lacko xlacko1@fi.muni.cz

2022-09-16

Obsah

Úvod

Úvod: Informácie o predmete

PB173 Tematické programování

- bez prednášok
- každá skupina má vlastné pravidlá
- možný opakovaný zápis

Úvod: Informácie o predmete

PB173/linux Systémové programování Linux

- programovanie v používateľskom priestore Linux Kernel
- ukážky použití rôznych rozhraní a súvisiacich knižníc
  - POSIX C Library, GNU C Library
  - POSIX Threads Library
  - rozhrania iných systémov, ktoré Linux implementuje
- programovací jazyk **C99** alebo **C11**

Úvod: Predpoklady

- znalosti predmetu *PB071 Nízkoúrovňové programování*
  - programovanie v jazyku C
  - používanie ladiacich nástrojov (`valgrind`, debugger)
- základné používateľské znalosti s Linux
- základy Gitu

Úvod: Materiály

Priebežne pridávané do projektu v GitLabe (viď `handbook.pdf`).

- **`exNNt-*`** komentované ukážky
- **`exNNp-*`** príklady na cvičenie
- **`exNNx-*`** doplňujúce alebo pokročilé ukážky
- **`hwNNa-*, ...`** základné domáce úlohy
- **`hwNNx-*, ...`** náročnejšie domáce úlohy

Úvod: Študijné materiály

- PDF prezentácia v študijných materiáloch v IS MU
- materiály k cvičeniu vo verejnom Git repozitári

Úvod: Bodovanie

10 b	jedna **základná** úloha z každého týždňa podľa vlastného výberu
5 b	každá *ďalšia* vypracovaná základná úloha z toho istého týždňa
10 - 30 b	každá **ťažká** domáca úloha, bodovanie bude upresnené v zadaní úlohy
0 - 2 b	vypracovanie príkladu na projektor, aktivita na hodine atď.

Úvod: Domáce úlohy

Jadrom hodnotenia sú domáce úlohy:

- Koniec odovzdávania **dva týždne** po zverejnení zadania.
- Úlohy sa odovzdávajú ako *Merge Request* v repozitári.
- Jedno opravné odovzdanie.

Úvod: Hodnotenie

Na úspešné hodnotenie je treba splniť nasledujúce podmienky:

1. Nanajvýš **dve** neospravedlnené neúčasti na seminári.
2. Získať aspoň **80 bodov**.
   Ekvivalent plných 8 z 12 základných úloh.

Náhrada neúčasti

Neospravedlnenú neúčasť si môžete nahradiť vypracovaním ťažkej úlohy z toho istého bloku za 0 bodov.

Ak budete túto možnosť chcieť využiť, dajte mi najprv vedieť.

Vývoj v prostredí GNU/Linux

The Linux philosophy is „Laugh in the face of danger.“
Oops. Wrong One. „Do it yourself.“ Yes, that's it.
— Linus Torvalds linux.dev.kernel newsgroup

GNU/Linux: Rozcvička

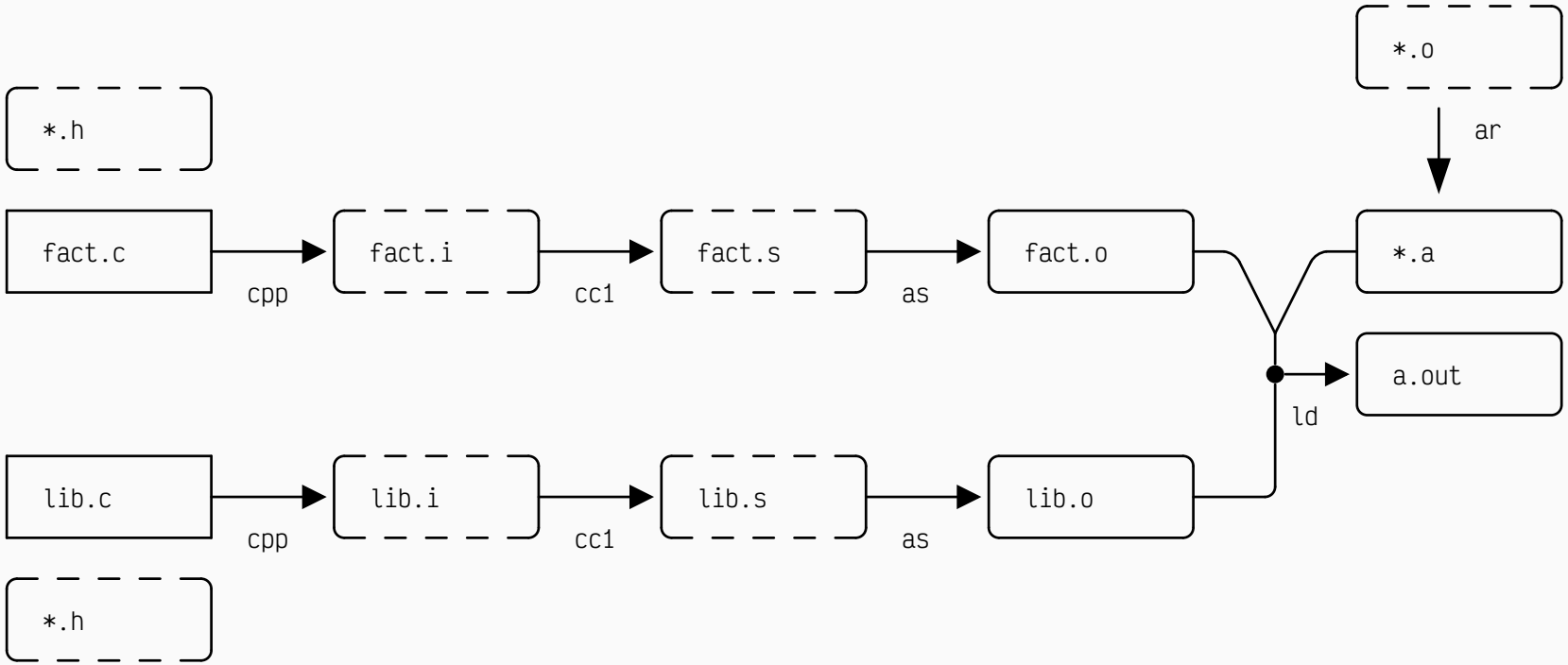
GNU/Linux: GCC

GNU Compiler Collection

- štandardný prekladač na dnešných UNIX systémoch
- dostupný i pre ďalšie platformy
- *front-endy* pre rôzne jazyky
  (C, C++, Fortran, Ada, ...)
- behové knižnice

GNU/Linux: GCC: Preklad programu

```
$ gcc fact.c lib.c
```



GNU/Linux: GCC: Fázy prekladu

`gcc` spúšťa jednotlivé časti prekladu samo.
Môžeme však preklad „zastaviť“ na niektorom kroku:

- **`-E`** až po preprocesor
- **`-S`** až po prekladač
- **`-c`** až po assembler

Pre pripomenutie

`-o FILE` uloží výstup do súboru `FILE`

GNU/Linux: GCC: Ďalšie prepínače

`-g`, `-pg` ladiace resp. profilovacie informácie
viď `gdb(1)`, `gprof(1)`

`-O0`, ..., `-O3` úroveň optimalizácií

`-Og` zapne len optimalizácie, ktoré nenarušia ladenie

`-Os` ako `-O2` okrem optimalizácií, ktoré zväčšujú výsledný program

`-Ofast` ako `-O3` a navyše nejaké ďalšie *potenciálne nebezpečné* optimalizácie

GNU/Linux: GCC: Štandard

`-std=STD` zapne konkrétny štandard jazyka

`-pedantic` vypne neštandardné rozšírenia prekladača

Vypne všetky symboly, ktoré nie sú súčasťou vybraného štandardu. Ak ich chceme používať, je nutné ich zapnúť:

- makrom v zdrojovom kóde

```
#define _GNU_SOURCE
#include <dlfcn.h>
```

- prepínačom pri kompilácii

```
$ gcc ... -D_POSIX_C_SOURCE=200809L ...
```

Viď `man feature_test_macros`

GNU/Linux: Make

Nástroj na automatické generovanie súborov z iných súborov. Okrem programov môže vytvárať prakticky čokoľvek.

```
$ make [-f FILE]
```

Po spustení hľadá súbory `Makefile` alebo `makefile`, z ktorých sa snaží prečítať recepty a spustí ich.

GNU/Linux: Make: Syntax

Syntax súborov:

- premenné (oficiálne „makrá“)

  ```
  OBJECTS = main.o utils.o
  ```

- pravidlá

  ```
  target: dependencies
      command
      ...
  ```

  Pozor, `command` musí byť odsadené `\t`!

- komentáre začínajú #

GNU/Linux: Make: Ciele

Make spúšťa ciele zadané na príkazovom riadku.
Ak nie sú zadané žiadne, vyrobí **prvý** cieľ. Ten sa typicky volá `all`.

Konvenčné ciele v Makefile:

`all` Cieľ, ktorý vymenúva závislosti, ktoré sa majú vyrobiť pri bežnom spustení `make`.

`test` Spustí priložené testy.

`install` Skopíruje vytvorený program do nejakého systémového umiestnenia.

`clean` Zmaže dočasné súbory a medzikroky kompilácie.

GNU/Linux: Make: Parametre a prepínače

`-n` Príkazy vypíše, ale nespustí.

`-k` Snaží sa vyrobiť čo najviac cieľov, neskončí pri prvej chybe.

`-p` Vypíše zabudovanú databázu pravidiel.

`-f FILE` Použije pravidlá zo súboru `FILE` namiesto `Makefile` alebo `makefile`.

`VAR=VALUE` Nastaví premennú z príkazového riadka.

Všetky ostatné parametre (bez -) sú ciele.

GNU/Linux: Make: Automatické premenné

Okrem vlastných premenných vytvára Make v kontexte pravidiel aj niekoľko užitočných makier:

$@ cieľ pravidla

$< prvý reťazec v zozname závislostí

$^ zoznam závislostí bez duplicít

$* cieľ pravidla bez prípony
koreň v implicitných pravidlách (GNU)

$+ (GNU) závislosti s opakovaním

$? (GNU) závislosti novšie než cieľ

GNU/Linux: Make: Príponové pravidlá

Vzory pravidiel podľa prípon súborov (prípony musia byť deklarované ako závislosti `.SUFFIXES`):

```
.SOURCE.TARGET:
    command
```

Príklad pravidla, ktoré má Make zabudované v sebe:

```
.SUFFIXES: .c .o


.c.o:
    $(CC) $(CFLAGS) -c $<
```

GNU/Linux: Make: Vzorové pravidlá (GNU)

Rozšírenie GNU na pohodlnejšiu syntax vzorov

```
%.o: %.c
    $(CC) $(CPPFLAGS) $(CFLAGS) -c $<

%: %.o
    $(CC) $(LDFLAGS) -o $@ $^ $(LDLIBS) $(LOADLIBES)
```

Podobné pravidlá má GNU Make už zabudované.

```
make -p -f/dev/null | grep -E -A3 '(%: %.o|%.o: %.c)$'
```

GNU/Linux: Make: Programy a prepínače

Niektoré premenné majú v Make význam daný POSIXom:

`CPP`, `CPPFLAGS` preprocesor a prepínače

`CC`, `CFLAGS` prekladač C a prepínače

`LDFLAGS` prepínače pre linker

`RM` prenositeľné mazanie súborov

`CXX`, `CXXFLAGS` **(GNU)** prekladač C++ a prepínače

`LOADLIBES`, `LDLIBS` **(GNU)** starý a nový zoznam knižníc

Meňte len `*FLAGS` a `LDLIBS`. Nástroje nechajte na prostredie.

Záver

Zdroje a odkazy

- manuálové stránky
  - `gcc(1)`
  - `make(1p)`
- GNU Make