

# GNU toolchain workflow

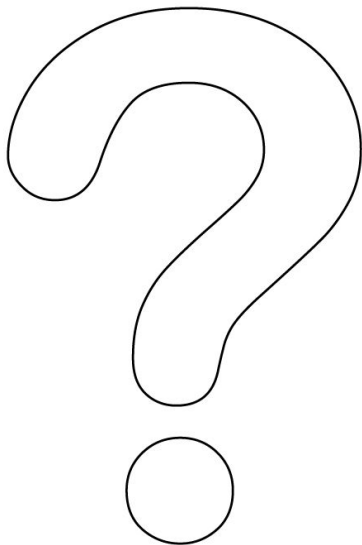
Projekt

FER, 2017

# Sadržaj

1. gcc prevoditelj i OS Linux
2. ostali alati za razvoj u OS Linux
3. primjeri i problemi

Što je potrebno za razvoj aplikacije?



# Potrebni alati

- ▶ gcc - prevoditelj
- ▶ ld - poveziavač
- ▶ strace - ispis poziva operacijskom sustavu
- ▶ objdump - sadržaj slike aplikacije
- ▶ hexdump - sadržaj binarnih datoteka
- ▶ Make - izgradnja sustava
- ▶ još nešto?

# Primjer 1

Napisati program u programskom jeziku C koji će ispisati “Glavni program” na standardni izlaz. Odrediti naredbu koju je potrebno pokrenuti iz ljuške operacijskog sustava Linux za prevođenje izvornog koda programa. Izvorni kod programa nalazi se u datoteci “glavni.c”.

# Pitanja 1

- ▶ Kako izgleda struktura prevedenog koda?
- ▶ Što se sve događa pozivom gcc naredbe?
- ▶ Kako se zapravo pokreće program?
- ▶ obj vs \*.elf vs \*.bin?
- ▶ Compiler, linker, assembler?
- ▶ Koja je razlika između generiranog binarnog oblika datoteke za računalo i Arduino?

## Primjer 2

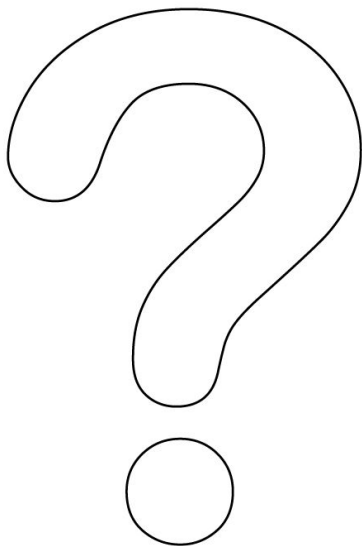
Napisati program u programskom jeziku C koji će na temelju ulaznog parametra koji je cijeli broj izračunati njegov kvadrat i ispisati ga na standardni izlaz. Datoteke izvornog koda (\*.c) se pritom moraju nalaziti u “src” direktoriju, a datoteke zaglavlja “inc” direktoriju. Izvršne datoteke “\*.elf” moraju biti u “exe” direktoriju. Napisati skriptu za prevođenje programa. Neka datoteka `main.c` sadrži funkciju `main`, `calc.c` neka sadrži funkciju `square`, a `help.c` funkciju `print_int`. Datoteka zaglavlja `modules.h` neka sadrži prototipe potrebnih funkcija.

## Pitanja 2

- ▶ Koju će grešku javiti prevoditelj, a koju poveziavač?
- ▶ Koja je razlika između `""` i `<>` prilikom `include` pretprocesorske direktive?
- ▶ Kako eksplicitno pozvati pretprocesor?
- ▶ Kako se predaju argumenti programu?



Mane ovakvog pristupa izgradnje programa?



# Mane ovakvog pristupa

- ▶ dodavanje datoteka
- ▶ višestruki posao u slučaju jedne promjene
- ▶ velika je vjerojatnost pogreške

## Primjer 3

Programsko rješenje iz prethodnog primjera izgraditi korištenjem Make alata. Odvojiti postupak povezivanja od prevođenja. Osigurati ponovnu izgradnju projekta u slučaju da se promjeni neka od datoteka izvornog koda ili Makefile. Osigurati perzistenciju pomoćnih datoteka u svrhu brže izgradnje programa.

## Primjer 4

Makefile datoteku iz prethodnog primjera modificirati tako da je moguće dodavati nove datoteke bez promjene Makefile datoteke.

# Zaključak

- ▶ nije vam potreban IDE, GNU je IDE