

## Vývoj programů.

Programovací nástroje. Aplikační programové rozhraní API. Standard POSIX. Vytvoření spustitelného souboru, použití kompilátoru a linkeru. Statické a dynamické (sdílené) knihovny. Ladění programů.



České vysoké učení technické Fakulta elektrotechnická

# Požadavky na programy

- ◆ **Efektivnost/výkonnost** - čím méně zdrojů program k dosažení cíle použije, tím lépe. Program musí také se zdroji korektně nakládat: např. mazat dočasné soubory či minimalizovat úniky paměti.
- ◆ **Spolehlivost** - jak často jsou výsledky běhu programu správné. (správnost algoritmu, chyby při programování)
- ◆ **Robustnost** - jak se program vypořádá s chybami v okolí. (vstupní dat, nedostupnost potřebných prostředků)
- ◆ **Použitelnost** - uživatelské rozhraní budiž jasné, intuitivní, konzistentní a kompletní.
- ◆ **Portovatelnost** - na kolik hardwarových platform a operačních systémů je program přenositelný
- ◆ **Spravovatelnost** - jednoduchá modifikovatelnost v budoucnosti. Opravy (bug and security holes) a doplnění funkcí

# Volba jazyka

- ◆ Kompilátor vs. interpret
- ◆ Úroveň jazyka
- ◆ Vhodnost jazyka pro danou úlohu

# Programovací nástroje

- ◆ programátorské editory
    - ◆ + syntaktická kontrola, našeptávač, zvýraznění (obarvení) kódu
  - ◆ kompilátory/interprety zvolených jazyků
  - ◆ debuggery
  - ◆ profilovací nástroje
  - ◆ návrhář grafického rozhraní
  - ◆ podporu verzování (CVS)
  - ◆ nástroje pro práci s komentářem a vytváření dokumentace
- 
- ◆ integrovaná vývojová prostředí

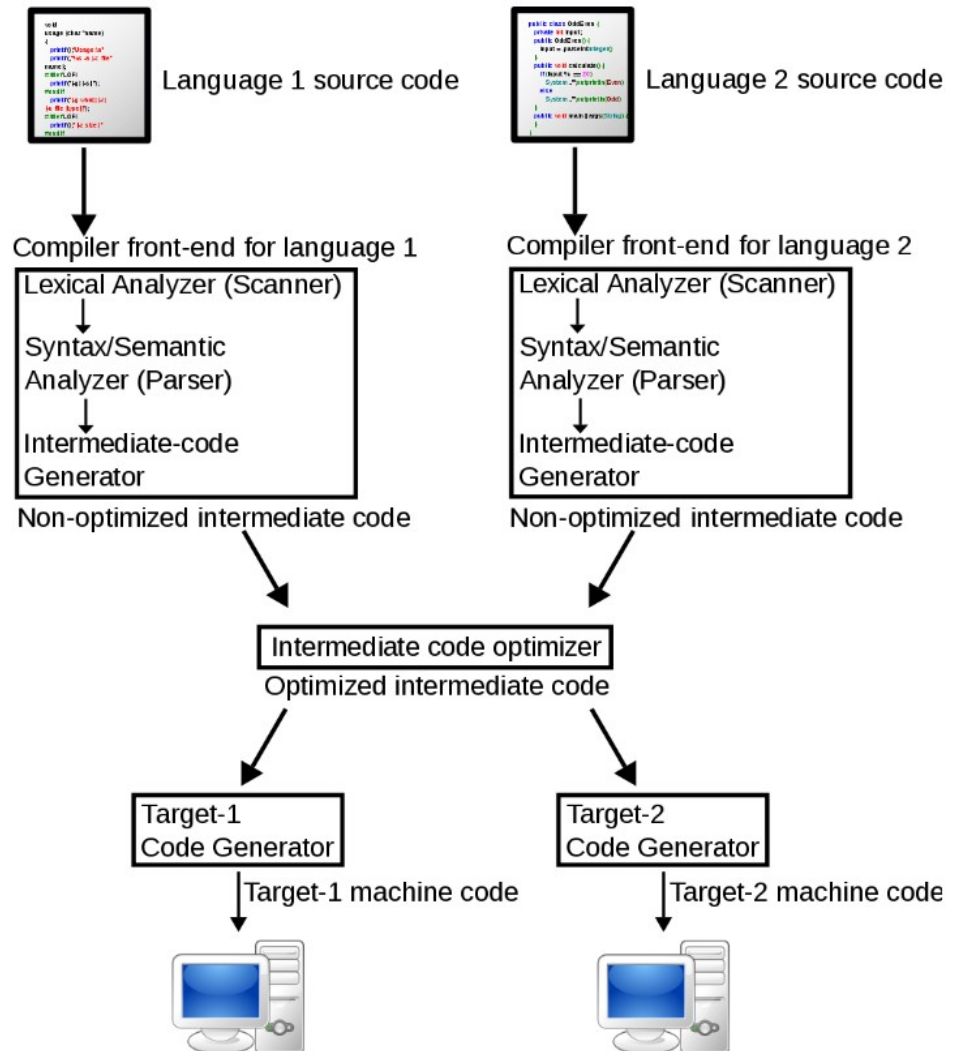
# API - POSIX

- <http://www.opengroup.org/onlinepubs/9699919799/>
- IEEE 1003 a ISO/IEC 9945
- Single Unix Specification (SUS), Linux Standard Base

# Kompilace

## ◆ Proč?

- Potřebujeme program upravit => zásah do zdrojových kódů.
- Neexistuje binární balíček pro naši distribuci.
- Chceme provést překlad na míru.
- Potřebujeme balíček pro nestandardní architekturu.
- Balíčky, které máme, nejsou s knihovnami v naší distribuci kompatibilní.



# Linkování

- větší projekty se píší po částech a ty se překládají jednotlivě - object moduly
- knihovny - není třeba tvořit již hotové
- vytvořené objekty je třeba pospojovat => linker
- staticky nebo dynamicky?

```
/usr/bin/ld --eh-frame-hdr -m elf_x86_64 -dynamic-linker /lib64/ld-linux-x86-64.so.2 /usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../lib64/crt1.o /usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../lib64/crti.o /usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/crtbegin.o -L/usr/lib64/gcc/x86_64-slamd64-linux/4.2.3 -L/usr/lib/gcc/x86_64-slamd64-linux/4.2.3 -L/usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../lib64 -L/lib/../../lib64 -L/usr/lib/../../lib64 -L/usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../x86_64-slamd64-linux/lib -L/usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../hello.o -lgcc --as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/crtend.o /usr/lib64/gcc/x86_64-slamd64-linux/4.2.3/../../../../lib64/crtn.o
```

# make + makefile

- ◆ zajistí kompilaci, linkování, ...

```
mc - ~/archive/software/peek_analysis_2
makefile [----] 0 L:[ 1+ 0 1/120] *(0 /4431b)= a 97 0x61
all: caod_diff_caod caod_diffv_d_caod caod_select_by_caoli caod_sum_caod \
caoli_histogram_caod caolie_det_peek_caod caoxy_concatenate_caod \
d_average_caod d_max_caod d_min_caod d_mgeoa_caod conv_aoxyh01_aoxy

clean:
rm -f *.o

mrproper:
rm -f *.o; rm -f caod_diff_caod caod_diffv_d_caod caod_select_by_caoli \
caod_sum_caod caoli_histogram_caod caolie_det_peek_caod \
caoxy_concatenate_caod d_average_caod d_max_caod d_min_caod \
d_mgeoa_caod conv_aoxyh01_aoxy

caod_diff_caod.o: caod_diff_caod.cpp td_caod.h dbg.h caod_fget.h i_fput_caod.h
gcc -c caod_diff_caod.cpp -o caod_diff_caod.o

caod_diff_caod: caod_diff_caod.o dbg.o caod_fget.o i_fput_caod.o
gcc caod_diff_caod.o dbg.o caod_fget.o i_fput_caod.o -o caod_diff_caod

caod_diffv_d_caod.o: caod_diffv_d_caod.cpp caod_fget.h i_fput_caod.h dbg.h
gcc -c caod_diffv_d_caod.cpp -o caod_diffv_d_caod.o

caod_diffv_d_caod: caod_diffv_d_caod.o caod_fget.o i_fput_caod.o dbg.o
gcc caod_diffv_d_caod.o dbg.o caod_fget.o i_fput_caod.o -o caod_diffv_d_caod

caod_fget.o: caod_fget.cpp dbg.h td_caod.h
gcc -c -O2 caod_fget.cpp -o caod_fget.o

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

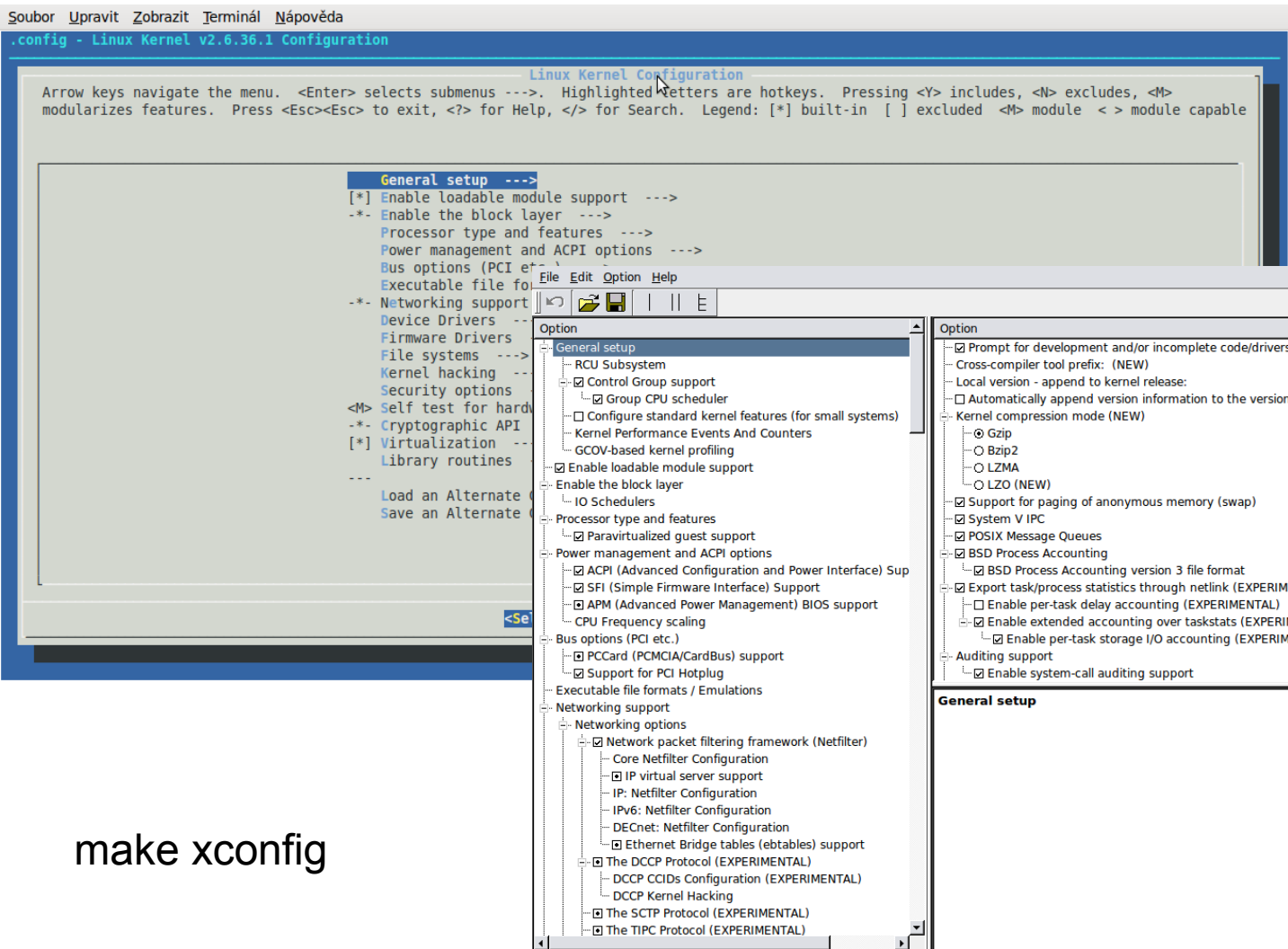


# Instalační trojkombinace

- ◆ **configure** - umožní nastavení požadavků (často jen jako přepínače z příkazového řádku), otestuje cílový systém a podle něj připraví následnou kompilaci, vyžádá si doinstalování chybějících programů potřebných pro překládaný program.
- ◆ **make** - přeloží a slinkuje nový program
- ◆ **make install (checkinstall)** - nainstaluje přeloženou aplikaci (kam?, a jak se odinstaluje?)

# Kde se vzalo jádro? - konfigurace

make menuconfig



make xconfig

## Kde se vzalo jádro? - kompilace

- ◆ make
- ◆ make bzImage
- ◆ make modules
- ◆ make modules\_install
  
- ◆ make mrproper
- ◆ make clean

# Ladění

- ◆ nástroje:
  - ◆ strace,
  - ◆ gdb,
  - ◆ valgrind
- ◆ ladící kód jako součást programu

# Distribuce

- ◆ distribuce - víceméně kompletní sada programového vybavení
  - ◆ binární/zdrojové
  - ◆ live
  - ◆ komerční/nekomerční
  - ◆ příklady: debian, gentoo, redhat, slackware, suse, ubuntu
- ◆ balíčkovací systémy
  - ◆ řeší korektní instalaci a odinstalaci jednotlivých programů
  - ◆ řeší závislosti mezi jednotlivými programy - balíčky