



Tomáš Čejka

cejkat@cesnet.cz

Autotools

co nám usnadňuje práci

24. 3. 2015

- Ručně: `gcc -Wall -g -o program *.c`

- ... (vlastní skripty)

- Jednoduchý Makefile:

```
echo "main: main.c mod1.c" > Makefile; make
```

Nevýhody ručních Makefile souborů?

- Závislosti mezi moduly
- Distribuce
- Instalace

Vše je potřeba řešit ručně...

Pro větší projekty může být dost pracné...

K Makefile se vrátíme později.

Kompilace nestačí...

Jak otestovat dostupnost závislostí?

Jak vypnout/zapnout volitelné součásti projektu?

Kam se má projekt instalovat?

Je to shell skript, který dokáže:

- otestovat dostupnost překladače, knihoven, hlavičkových souborů,
- umožňuje uživateli nastavit projekt

`configure` lze vygenerovat ze šablony `configure.ac` pomocí `autoreconf(1)`.

Makefile + configure?

Aneb jak to spojit dohromady.

Workflow (důležitý slajd)



autoreconf(1)		
Vytvoří configure a soubory .in; interně spouští např. automake(1)		

Workflow (důležitý slajd)



autoreconf(1)	configure	
Vytvoří configure a soubory .in; interně spouští např. automake(1)	Vytvoří soubory Makefile a zpracuje všechny .in soubory, umožňuje uživateli nakonfigurovat projekt	

Workflow (důležitý slajd)



autoreconf(1)	configure	make(1)
Vytvoří configure a soubory .in; interně spouští např. automake(1)	Vytvoří soubory Makefile a zpracuje všechny .in soubory, umožňuje uživateli nakonfigurovat projekt	Spustí překlad/vytvoření distribučního balíku/instalaci/kontrolu

Jak to vypadá v praxi?

Potřebujeme dodat 2 typy souborů:

`configure.ac` a `Makefile.am`

Jak to vypadá v praxi?

Potřebujeme dodat 2 typy souborů:

`configure.ac` a `Makefile.am`

Šablona: `configure.ac`

- Je možné ji automaticky vygenerovat pomocí `autoscan(1)`
- Vznikne `configure.scan`: `mv configure.{scan,ac}`
- Nutná ruční úprava! (např. pro spouštění `automake(1)` nutno dodat `AM_INIT_AUTOMAKE()`)

Jak to vypadá v praxi?

Potřebujeme dodat 2 typy souborů:

`configure.ac` a `Makefile.am`

Šablona: `configure.ac`

- Je možné ji automaticky vygenerovat pomocí `autoscan(1)`
- Vznikne `configure.scan`: `mv configure.{scan,ac}`
- Nutná ruční úprava! (např. pro spouštění `automake(1)` nutno dodat `AM_INIT_AUTOMAKE()`)

Šablona: `Makefile.am`

- Nutno napsat ručně... (ale stačí jen pár řádků)
- Obsahuje seznam programů, knihoven, skriptů, veřejných hlavičkových souborů...
- ... A jejich zdrojů

```
cat > Makefile.am <<KONEC
bin_PROGRAMS=muj-program
muj_program_SOURCES=main.c mod1.c mod1.h
muj_program_LDADD=-ltrap -lm
muj_program_CPPFLAGS=-DMAKRO_CPP
KONEC
```

Výhody použití Autotools? I

Statistika:

Kolik jsme **napsali** a co se vygenerovalo.

Řádky	Slova	Znaky	
1	1	12	./Makefile.am
2	4	64	src/Makefile.am
24	53	567	configure.ac
550	1979	16980	src/Makefile
745	2775	23978	./Makefile
5450	20384	160992	configure
6772	25196	202593	celkem

Výhody použití Autotools? II

Díky vygenerovaným souborům můžeme použít:
make, make install, make uninstall, make dist, make distcheck,
make ctags

Snadné rekurzivní iterování přes adresáře pomocí:

SUBDIRS=<podadresář>

Máme připraveno:
make check

- získejte:

`https://github.com/cejkato2/autotools-homework`

- získejte:
`https://github.com/cejkato2/autotools-homework`
- Vytvořte `configure.ac`, `Makefile.am` a vyzkoušejte :-)

- získejte:
`https://github.com/cejkato2/autotools-homework`
- Vytvořte `configure.ac`, `Makefile.am` a vyzkoušejte :-)
- V `README.md` je nápověda.

- získejte:
`https://github.com/cejkato2/autotools-homework`
- Vytvořte `configure.ac`, `Makefile.am` a vyzkoušejte :-)
- V `README.md` je nápověda.

- získejte:
`https://github.com/cejkato2/autotools-homework`
- Vytvořte `configure.ac`, `Makefile.am` a vyzkoušejte :-)
- V `README.md` je nápověda.

Dotazy na `cejkat@cesnet.cz`