

Poprvé s Make

Program **make** podporuje programátora při **překládání** zdrojových souborů a **sestavování** spustitelných programů; umožňuje efektivně spravovat i složité překladové kaskády. Programátor pomocí konfiguračního souboru **makefile** vyjádří **závislosti** mezi jednotlivými kroky (díličními překlady) k sestavení cílového programu a pak už jen cyklicky edituje zdrojové soubory a volá příkaz make. Ten nalezne a volá pouze ty kroky, které jsou pro nové sestavení programu nezbytné. V Linuxu je make běžně k dispozici ve verzi **GNU make**, do Windows se dostane např. spolu s GNU překladači nebo jako **Microsoft nmake** spolu s vývojovým prostředím Microsoft Visual Studio.

Ukázka souborů makefile s komentářem a jednou překladovou kaskádou vyjádřenou dvěma **pravidly**:

```
# gfortran v Linuxu (1 program s kaskádou)          # gfortran ve Windows
a.out: a.o                                          a.exe: a.o
      gfortran a.o                                  gfortran a.o
a.o: a.f90                                         a.o: a.f90
      gfortran -c a.f90                            gfortran -c a.f90
```

Pravidlo (**rule**) se skládá z úvodního popisu závislosti cílového souboru (**target**) na komponentách (**components**) a následujících řádků uvozených tabulátorem (**tab key**, při použití středníku lze pravidlo sloučit na jeden řádek bez tabulátoru) s příkazy, kterými lze cílový soubor z komponent vytvořit. Make spuštěný bez argumentu testuje první pravidlo v makefile, jinak testuje pravidla uvedená jako argumenty; jsou-li komponenty testovaného pravidla cílem jiných pravidel, jsou nejprve testována tato pravidla. Pokud make shledá, že cílový soubor neexistuje nebo je starší než komponenty, provede příkazy testovaného pravidla, jinak nikoliv (vypíše, že cíl „is up to date“). Při vytváření makefiles se běžně používají proměnné neboli makra (**macros**), automatické proměnné, pravidla bez závislostí (**phony rules**), **prefixy** příkazů, implicitní překladová pravidla (**implicit rules**), vzorová pravidla (**pattern rules**) ad. Syntaxe makefiles bývá kritizována pro zastaralost a nedostatečnost, systém make je však dobře dostupný.

Ukázka: definice a použití maker v pravidle, automatická proměnná **\$\$** pro aktuální cíl, sloučení pravidla na jeden řádek (vhodné při nastavení editoru na automatickou konverzi tabulátorů na mezery; ve Windows lze i tehdy tabulátor vytvořit klávesovou kombinací Alt+009). Překládá se jednokrokově, bez kaskády:

```
# gfortran v Linuxu i Windows (1 program bez kaskády)
P=a
EXT=out      # Linux: EXT=out; Windows: EXT=exe
FC=gfortran
FFLAGS=-O3 -march=native -fopenmp
$(P).$(EXT): $(P).f90; $(FC) $(FFLAGS) -o $$ $(P).f90
```

Volby gfortranu a přípony jeho mezisouborů se v Linuxu a Windows neliší, makefiles jsou až na příponu cílového souboru stejné. S Intel překladači ifort a ifx je to mezi Linuxem a Windows horší, např. v kaskádě při dvou zdrojových souborech tvořících jeden program:

```
# Intel v Linuxu (1 program, 2 soubory, kaskáda)          # Intel ve Windows
FC=ifx                                                    FC=ifx
FFLAGS=-O3 -heap-arrays -xHost -qopenmp                 FFLAGS=-O3 -heap-arrays -QxHost -Qopenmp -nologo
prg.out: mod.o prg.o; $(FC) $(FFLAGS) -o $$ mod.o prg.o  prg.exe: mod.obj prg.obj; $(FC) $(FFLAGS) -o $$ mod.obj prg.obj
mod.o: mod.f90; $(FC) $(FFLAGS) -c mod.f90              mod.obj: mod.f90; $(FC) $(FFLAGS) -c mod.f90
prg.o: prg.f90; $(FC) $(FFLAGS) -c prg.f90              prg.obj: prg.f90; $(FC) $(FFLAGS) -c prg.f90
```

Zvyklostní pravidlo **all** (zde pro podmíněný překlad a nepodmíněný běh dvou programů), pravidlo bez závislostí **clean** (pro mazání souborů), prefixy pro potlačení opisu příkazu (**@**) a ignorování chybového kódu příkazu (**-**):

```
# gfortran v Linuxu: make all; make clean (2 programy, bez kaskády)
all:      a.out a b.out b
a.out:   a.f90; @echo compiling $$; gfortran -o $$ a.f90
b.out:   b.f90; @echo compiling $$; gfortran -o $$ b.f90
a:       ; @./a.out
b:       ; @./b.out
clean:   @echo Cleaning...
         @-rm -f *.out *.exe *.o *.obj *.mod
```

Manuál: <https://www.gnu.org/software/make/manual/>