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 k sestavení cílového programu. Po aktualizaci zdrojových souborů pak make nalezne a volá jen ty kroky, které jsou pro nové sestavení nezbytné. V Linuxu je make běžně k dispozici ve verzi GNU make, do Windows se dostane buď v téže podobě 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:

```
# Linux
a.out: prg.o
        gfortran prg.o
prg.o: prg.f90
        gfortran -c prg.f90
```

```
# Windows
a: prg.o
        gfortran prg.o
prg.o: prg.f90
        gfortran -c prg.f90
```

Pravidlo (rule) se skládá z úvodního popisu závislosti cílového souboru (target) na zdrojových souborech (components) a následujících řádků (uvozených tabelátorem) s příkazy, kterými lze cílový soubor ze zdrojových souborů vytvořit.

Make spuštěný bez argumentu testuje první pravidlo v makefile, jinak testuje pravidla uvedená jako argumenty; jsou-li zdrojové soubory testovaného pravidla cílem jiných pravidel, jsou nejprve testována tato pravidla. Pokud make shledá, že cílový soubor neexistuje nebo je starší než zdrojové soubory, provede příkazy testovaného pravidla, jinak nikoliv (cíl „is up to date").

Při vytváření makefiles se pro zjednodušení (někdy „zjednodušení") běžně používají proměnné neboli makra (macros), interní makra, implicitní překladová pravidla (implicit rules), pravidla vázaná na přípony souborů (suffix rules), pravidla bez závislostí (phony rules), prefixy příkazů ad. Syntaxe makefiles bývá kritizována pro zastaralost a nedostatečnost, systém make je však všude možně dobře dostupný.

Další ukázky
Makra: definice a použití v pravidle, interní makro $@ pro aktuální cíl

```
P=prg
$(P): $(P).f90
        gfortran -o $@ $(P).f90
```

Příponová pravidla pro fortranské zdrojové soubory, sloučené řádky, interní makro $< pro (první) zdrojový soubor

```
FC=gfortran
.SUFFIXES: .f90 .f .for .o .obj
.f90.o:  ; $(FC) -c $<
.f.o:    ; $(FC) -c $<
.for.o:  ; $(FC) -c $<
.f90.obj: ; $(FC) -c $<
.f.obj:  ; $(FC) -c $<
.for.obj:; $(FC) -c $<
```

Do takto vybaveného makefile by stačilo přidat pravidlo

```
$(P): $(P).o ; $(FC) -o $@ $(P).o
```

a make by pro překlad sám vyhledal jeden ze souborů se jménem $(P) a příponou .f90, .f nebo .for.

Pravidlo bez závislostí a prefixy pro potlačení opisu příkazu (@) a pro ignorování chybového kódu příkazu (-):

```
clean:
        @echo Cleaning...
        @-rm -f a.out a.exe *.o *.obj *.mod
```

Volání: make clean.

L. H., 12. 12. 2015