Instalace GHDL a GTKWave

Richard Šusta, Katedra řídicí techniky, FEL, ČVUT v Praze

Verze 1.0 z 17. března 2024

Domovská stránka dokumentu: https://dcenet.fel.cvut.cz/edu/fpga/install.aspx

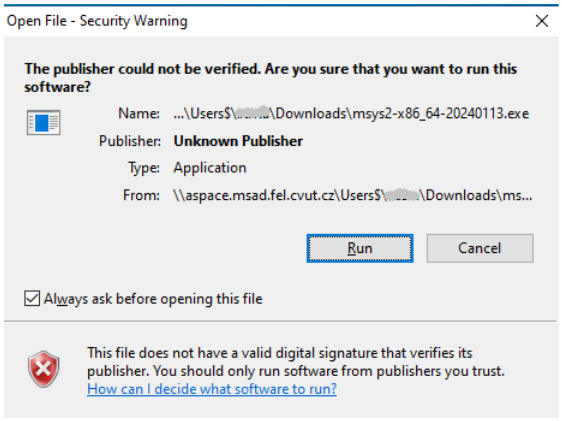
Obsah

O GHDL

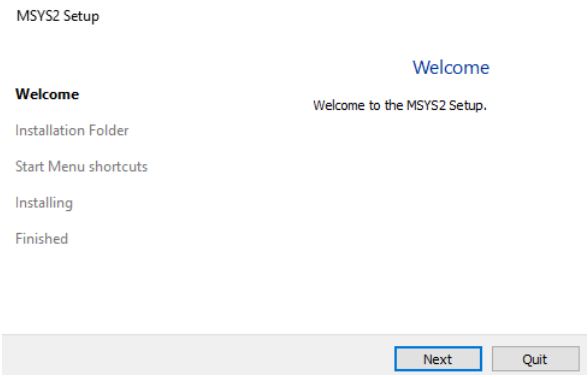
- GHDL kompiluje zdrojový kód VHDL na exe soubor spustitelný ve Windows počítači. Dle našich testů v něm proběhne cyklus překlad-simulace stejně rychle jako v profesionálním simulátoru ModelSim. Ten však vyžaduje znalost jeho vývojového prostředí, a dost odlišného od programovacích, a tvorbu projektu. Přesněji specifikuje chyby v kódu, ale jen tehdy, umí-li se s ním zacházet.
- GHDL se snadno spustí i z okna terminálu, které je součástí Visual Studio Code.
- Celá instalace GHDL s GTKWave potřebuje přibližně 2,7 GB, tedy o něco méně než ModelSim, a vznikla pro Linux. Nevyužívá registr Windows ani jejich systémové složky, a proto ji lze čistě odinstalovat.
- GHDL plně podporuje verze standardů IEEE 1076 VHDL z let 1987, 1993 a 2002 a hlavní rysy jeho revize z roku 2008.
- Běží v systémech Linux, Windows a Apple OS X a je open-source projekt, který má dodnes aktivní přispěvatele, viz https://github.com/ghdl/. Lze si bezplatně stáhnout binární distribuci nebo zdrojový kód a ten si zkompilovat na svém počítači.
- Instalace GHDL je jednoduchá v Linuxu, ale ve Windows složitější, protože v nich vyžaduje přidání MINGW, *Minimalist GNU for Windows*.
- **Pozor**! Existují také nativní GHDL Windows verze, ale všechny jsou zastaralé. Autoři je již léta nevytvářejí. Projekt GHDL nyní podporuje jedině **MSYS2**, *Minimal System*, tj. komponentu MinGW (*Minimalist GNU for Windows*) coby zavedenou platformu pro distribuci softwaru ve Windows.
- **Musíme tady začít s MSYS2.**

Instalace MSYS2

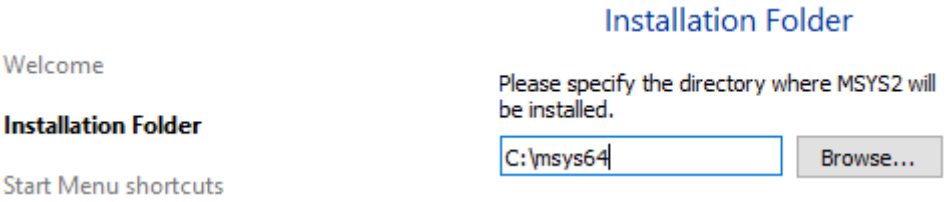
Přejděte na oficiální web projektu a stáhněte si instalační program. Jeho nejnovější verze byla msys2-x86_64-20240113.exe v době psaní tohoto dokumentu. Spusťte instalační program a potvrďte bezpečnostní varování:



V úvodním dialogu klikněte na tlačítko [Další].

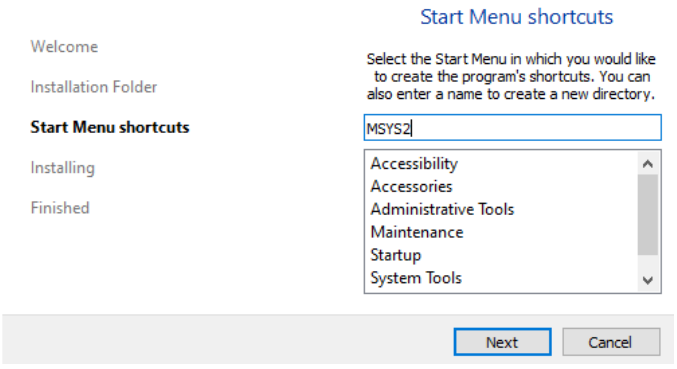


Zadejte požadovaný instalační adresář. Výchozí hodnota je **C:\msys64\.** Můžete zadat vlastní, ale rozhodně s **krátkou cestou mající ASCII** název, která leží na SSD nebo HD se systémem souborů **NTFS**. Bude-li dlouhá, brzy překročíte maximání povolenou délku. Další služby se budou totiž přidávat jako její podadresáře.

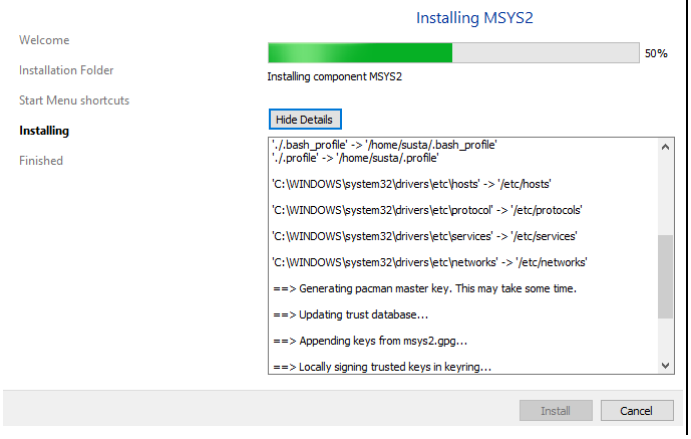


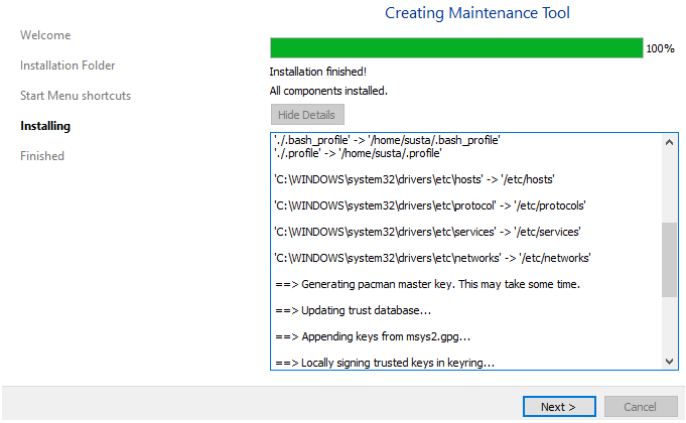
Cesta nesmí obsahovat diakritická znaménka, mezery, symlinky odkazující na jiný adresář a přidružené subst k virtuálním jednotkám. Podporovány nejsou ani síťové jednotky a svazky FAT.

Na následující obrazovce vyberte zástupce nabídky a spusťte instalaci pomocí tlačítka [Další].



Instalace bude hlásit všechny své akce a nakonec skončí.

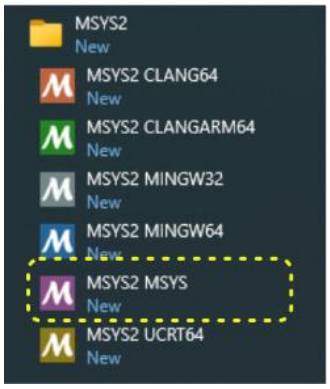




Pokud se zobrazí okno terminálu UCRT64, můžete jej zavřít. Nepotřebujeme ho.

Instalace GHDL

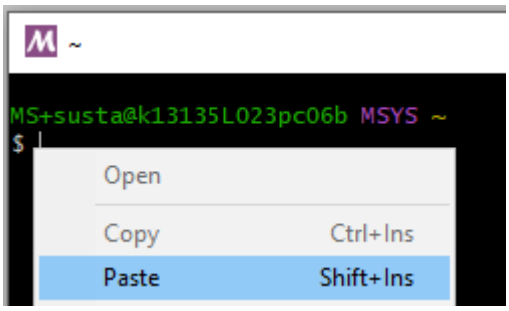
Pomocí MSYS2 můžeme již nainstalovat GHDL. Otevřete konzoli **MSYS2 MSYS** z nabídky *Start* systému Windows.

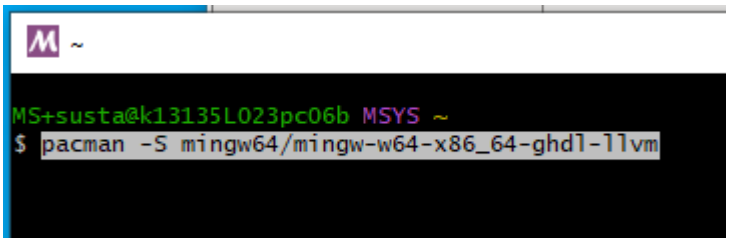


Pro instalaci 64bitové verze jazyka GHDL použijeme následující příkaz pro Pacman, linuxový správce paketů:

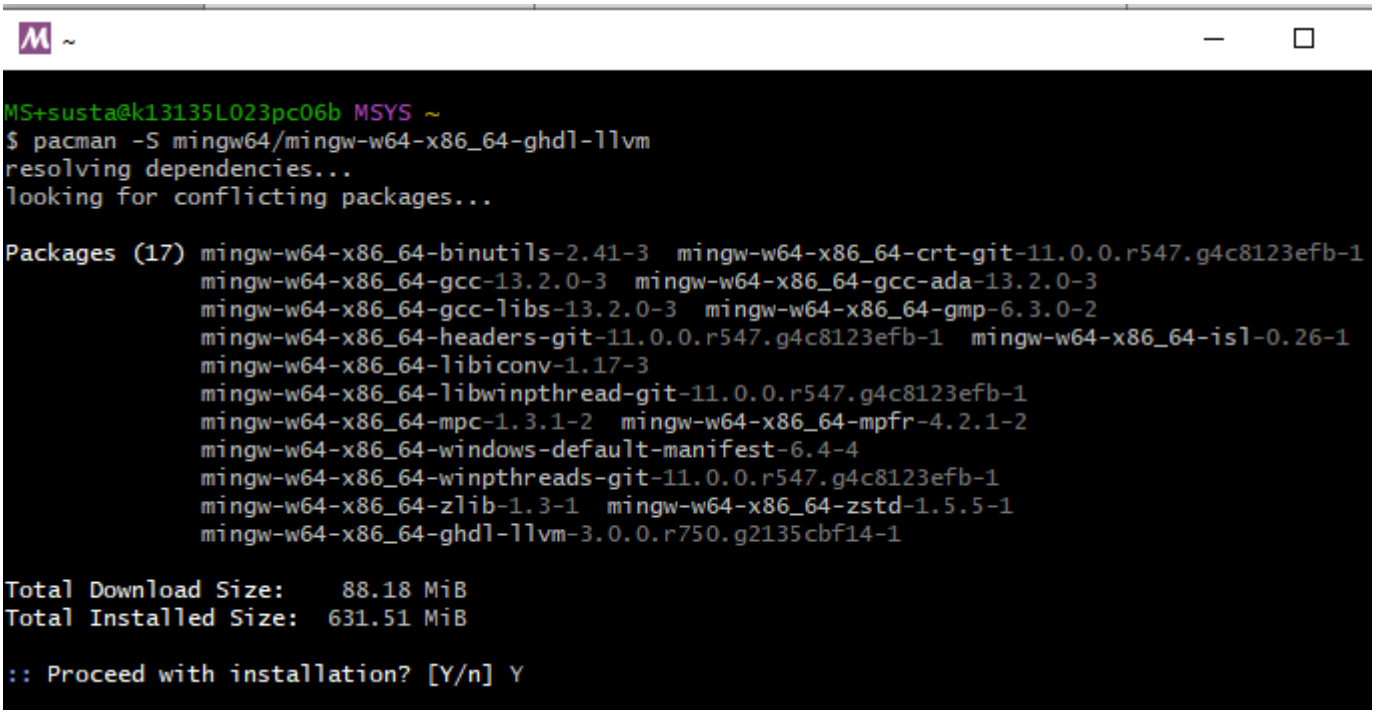
```
pacman -S mingw64/mingw-w64-x86_64-ghdl-llvm
```

Zkopírujte ji do příkazové konzole MSYS2 MSYS:

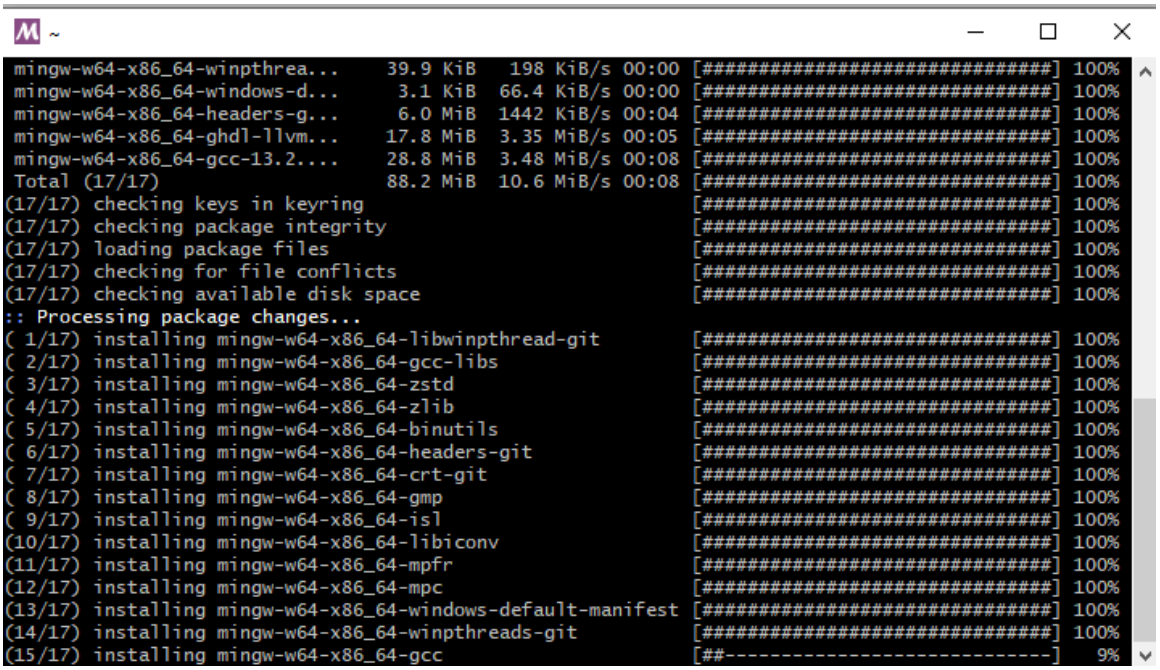




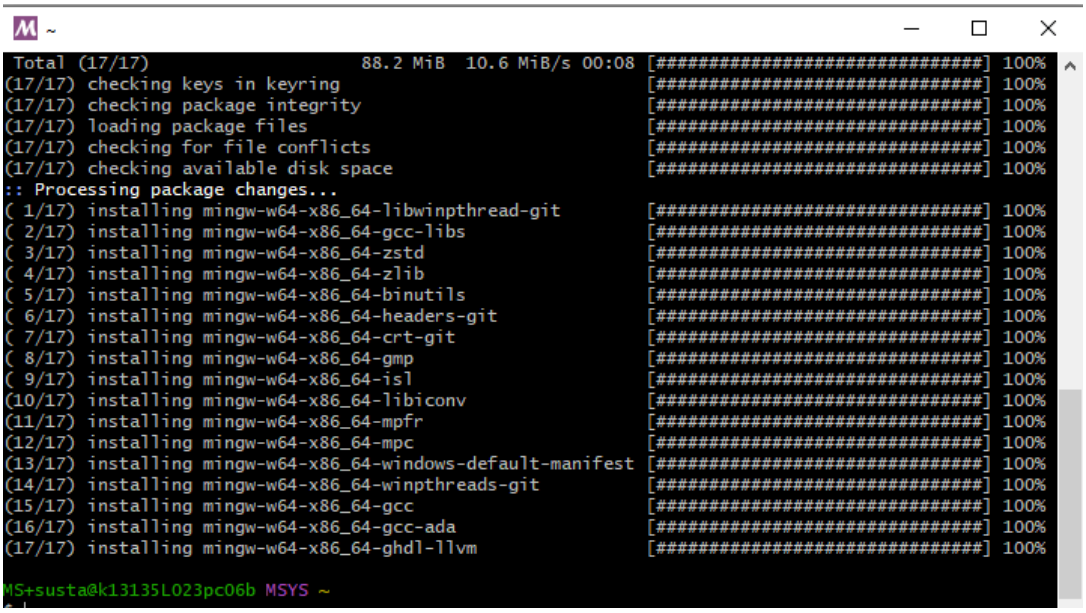
Pacman vyhledá potřebné změny. Jeho návrhy potvrďte zadáním Y.



Instalace bude opět hlásit všechny akce coby obecné chování linuxových programů.



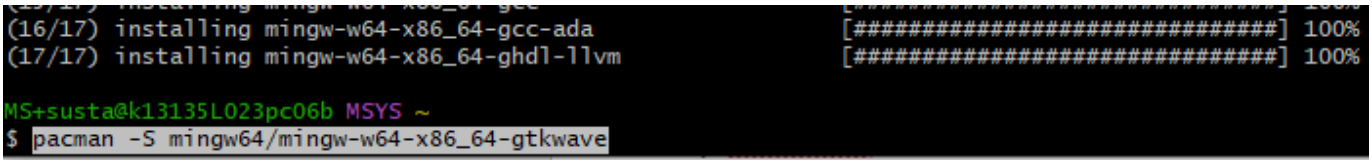
Nakonec ohlásí hotovo.



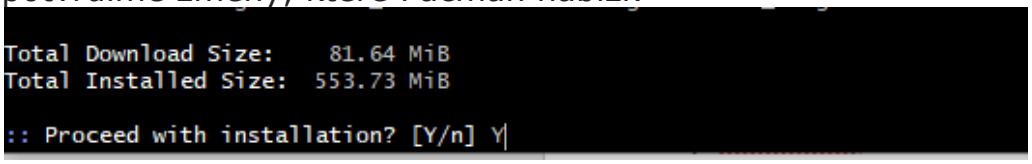
Instalace nástroje GtkWave

GTKWave potřebujeme k zobrazení signálů generovaných simulací GHDL. Zkopírujte následující řádek do okna MSYS2 a spusťte příkaz:

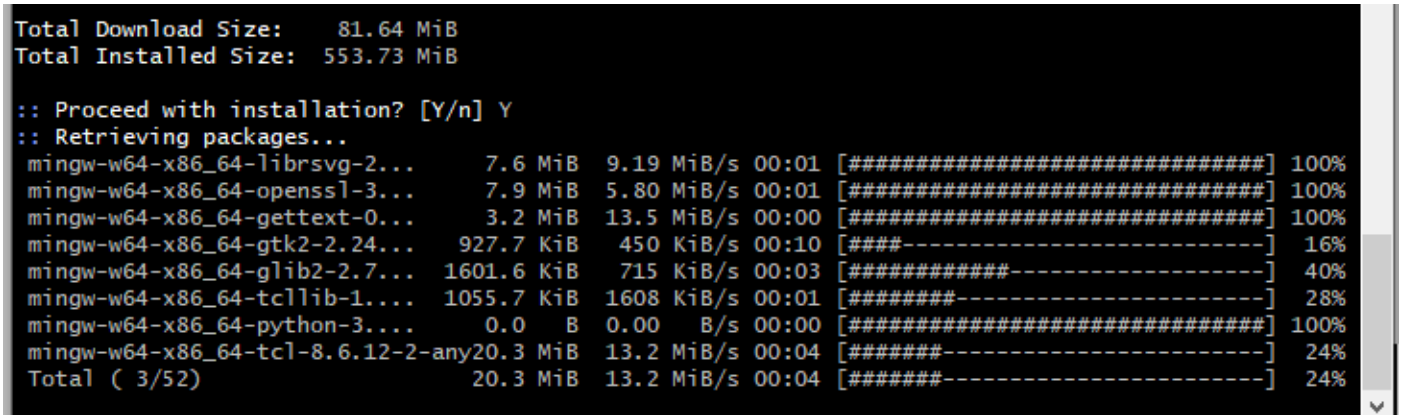
```
pacman -S mingw64/mingw-w64-x86_64-gtkwave
```



Zadáním Y potvrdíme změny, které Pacman nabízí:



Instalace GtkWave také hlásí všechny své akce:

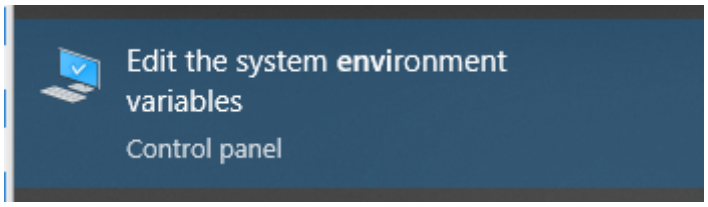


Až skončí, můžeme zavřít okno příkazového řádku MSYS.

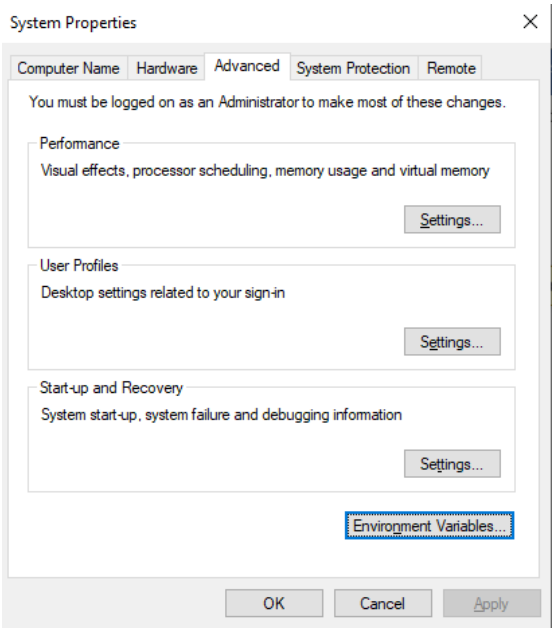
Nastavení prostředí

cestu ke ghdl.exe přidáme až na **konec** PATH Windows, což je robustnější.
V dávkových příkazech (*.bat) ji lze kdykoli dočasně přesunout na začátek PATH.

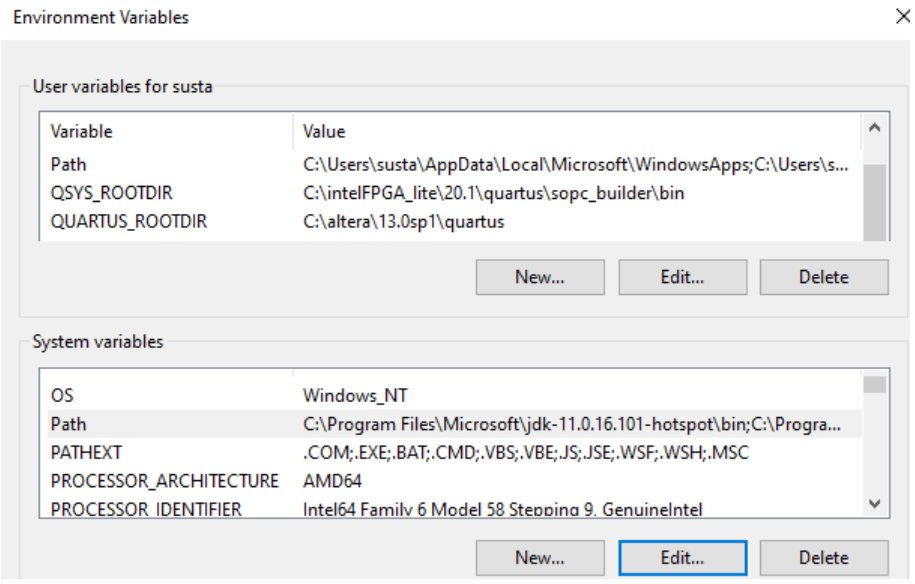
Stiskneme klávesu Windows a začneme psát slovo environment (prostředí). Po jeho prvních třech písmenech by měl systém Windows nabídnout volbu:



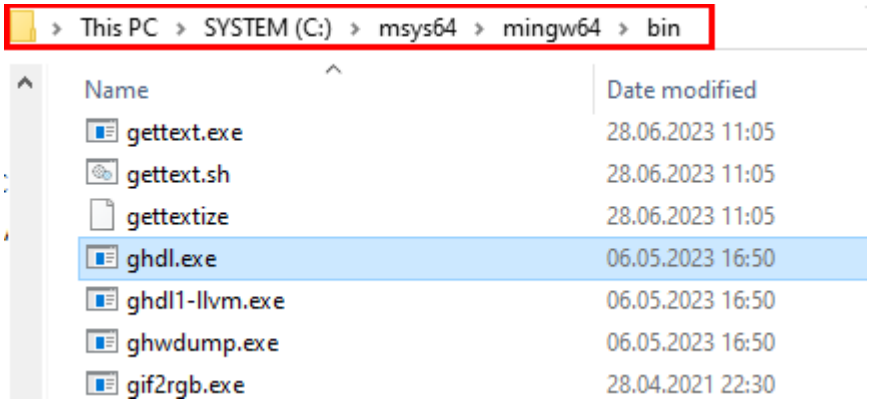
Vybereme jej a v okně Vlastnosti systému zvolíme [Proměnné prostředí...]



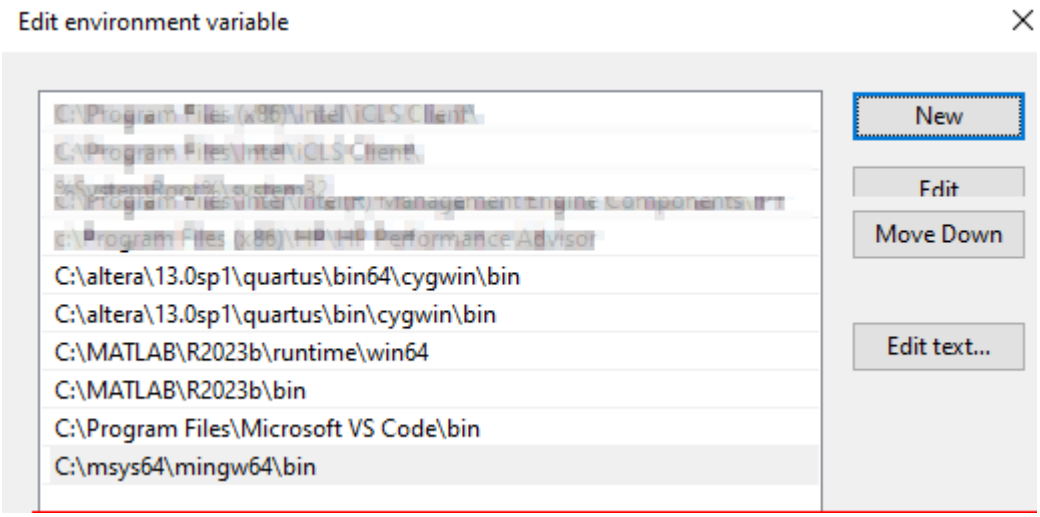
Poté editujeme Path cestu v systémových proměnných.



Zjistíme si umístění souboru ghdl.exe. Pokud jsme nezměnili výchozí nastavení, je:



Stiskem [NEW] vložíme cestu k němu až na **konec** PATH:

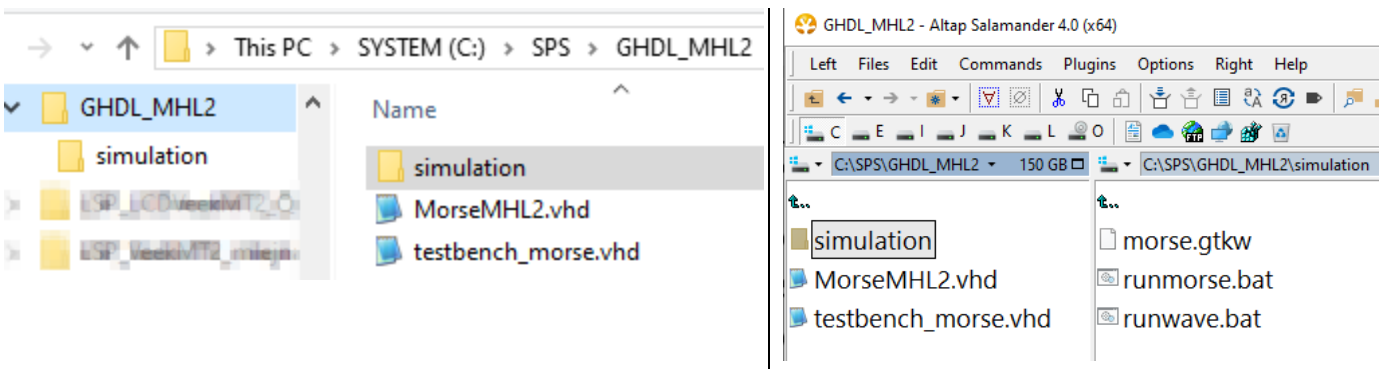


Příklad simulace v prostředí GHDL

Na počítači máme na instalovaný Quartus, který používá MinGW, ale jeho jinou verzi než GHDL, což není nic nového v instalačním pekle počítačové reality, v jejímž kotli není moudré míchat verze. Práve kvůli tomu jsme cestu přidali až na konec PATH!

Ze stejného důvodu nemůžeme jednoduše spustit *ghdl.exe* samostatně nebo použít makefile, ale musíme příkaz zahloubit do dávkového skriptu Windows, ve kterém můžeme dočasně přesunout cestu k nástrojům GHDL na začátek.

Předpokládejme, že jsme všechny potřebné soubory přiloženého příkladu umístili do **C:\SPS\GHDL_MHL2**



Morse Beacon potřebuje pouze *soubory MorseMHL2.vhd* a *testbench_morse.vhd* pro GHDL. *Testbench_morse* se objasní na přednáškách.

V podsložce simulace máme 2 dávkové soubory. První je **runmorse.bat:**

```
@ECHO ON
@SETLOCAL
@rem Temporary moving mingw64 to top
@set PATH=C:\msys64\mingw64\bin\;%PATH%
@rem Compile for VHDL-2008
@set GHDL_FLAGS=-fsynopsys --std=08
@if exist morse.vcd del morse.vcd
@rem Analyze
ghdl.exe -a %GHDL_FLAGS% ../MorseMHL2.vhd ../testbench_morse.vhd
IF ERRORLEVEL 1 GOTO BAT-END
@rem Create exe
ghdl.exe -e %GHDL_FLAGS% testbench_morse
IF ERRORLEVEL 1 GOTO BAT-END
@rem RUN simulation
ghdl.exe -r %GHDL_FLAGS% testbench_morse --vcd=morse.vcd --
wave=morse.ghw --stop-time=1us
:BAT-END
```

Jeho příkazy:

`ECHO ON` - vypiš všechny řádky, které nezačínají znakem @

`SETLOCAL` - spustí lokalizaci proměnných prostředí. Změny budou platné až do konce dávkového souboru nebo k dosažení odpovídajícího příkazu ENDLOCAL.

`rem` - následuje komentář až do konce řádku.

`set PATH` - dočasný přesun C:\msys64\mingw64\bin\ na začátek systémové Path.

`set GHDL_FLAGS` - nastaví parametry nutné k aktivaci podpory VHDL 2008.

`del morse.vcd` - odstraníme *morse.vcd,* pokud existuje. Jeho přítomnost bude signalizovat, že skript se dokončil bez chyby.

`ghdl.exe -a` analyzuje soubory VHDL v nadřízeném adresáři. **Všimněte si jejich pořadí**! Soubor *morseMHL2.vhd* se musí uvést před *testbench_morse.vhd*.

`IF ERRORLEVEL 1 GOTO BAT-END` - skočí se na konec, skončí-li předchozí příkaz chybou.

`ghdl.exe -e` vytvoří simulaci v souboru *testbench_morse.exe*.

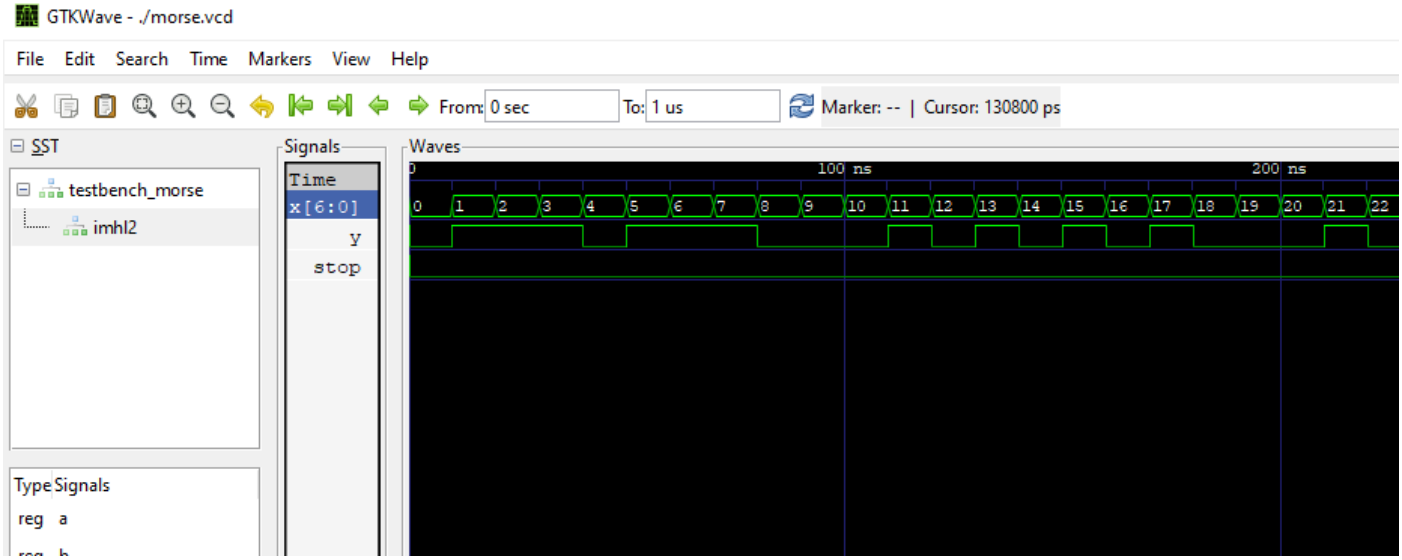
`ghdl.exe -r` spustí exe soubor a zaznamená průběhy signálů do *morse.vcd*.

Poznámka: Pokud chceme simulovat VHDL soubor s jiným názvem, pouze změníme MorseMHL2 na náš vlastní a analogickou úpravu učiníme i v testbench_morse.vhd.

Proběhne-li soubor *runmorse.bat* v pořádku, spustíme *runwave.bat*:

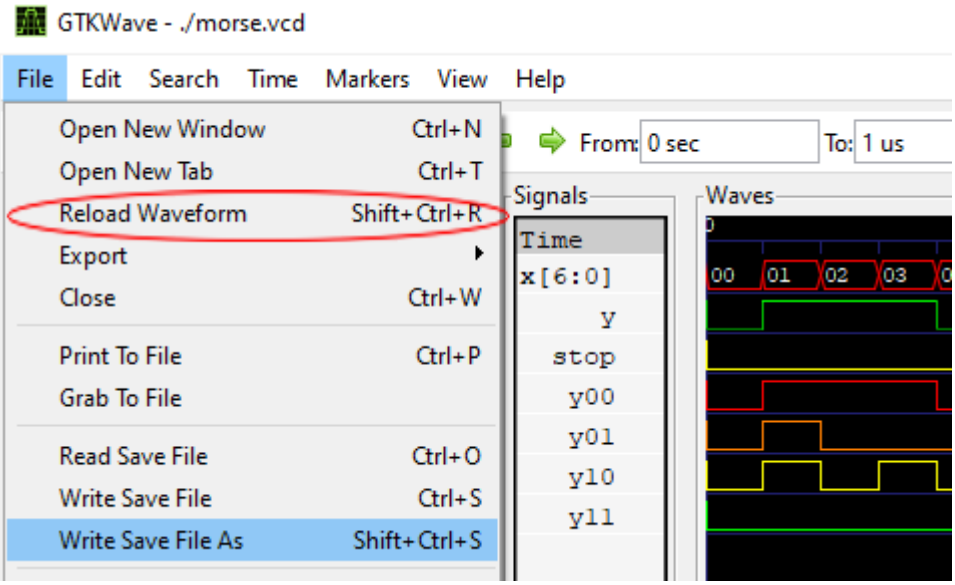
```
@IF exist ./morse.vcd ( start gtkwave ./morse.vcd ./morse.gtkw
 ) ELSE (
echo There's nothing to simulate. First, the runmorse.bat file must exit successfully.
pause )
```

Nejprve otestujeme úspěšné skončení *runmorse.bat,* a to testem, zda existuje soubor *morse.vcd.*
Poté spustíme samostatné okno příkazového řádku, v němž spustíme `gtkwave.exe`. Dávkový soubor *runwave.bat* tak běží dál až na konec a zavře se. Zabrání se tomu jen v případě chybové zprávy, kdy příkaz pauza vloží čekání na stisk nějaké klávesy.

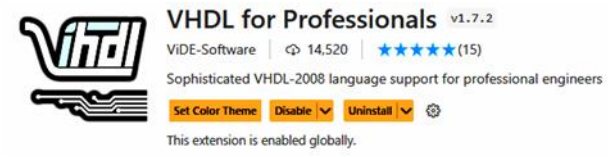


Program gtkwave obdržel při svém spuštění dva parametry:

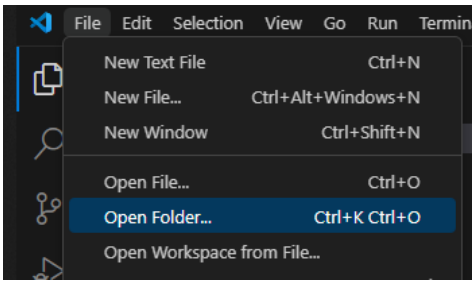
- První je *morse.vcd* (.vcd=Value Change Dump). Pokud následnou simulací vytvoříme nový *morse.vcd*, nemusíme znovu spustit *runwave.bat* pouze nový *.vcd načteme z GTKWave: **File->Reload Waveform**
- Druhým parametrem je soubor **morse.gtkw** (uložená relace GTKWave). Určuje signály, které se mají zobrazit, a jejich formáty. Můžeme si je přeskládat a uložit naši novou konfiguraci z GTKWave **File-> Write Save File As**.



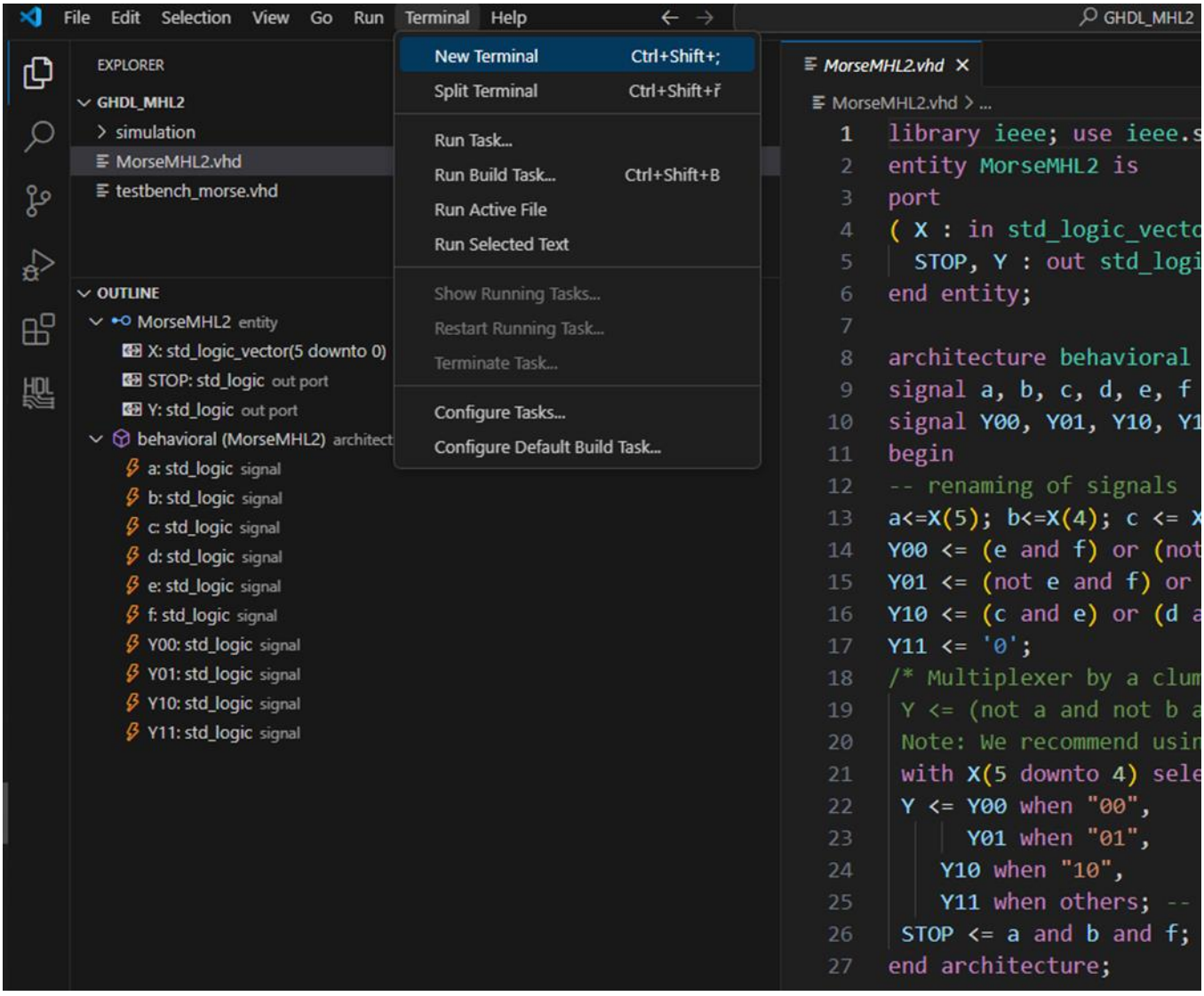
Dávkové skripty se spouštějí z okna příkazového řádku. Zde s výhodou používáme terminál VSC (Visual Studio Code), které má i mnoho rozšíření usnadňujících VHDL kódování. Následující obrázky byly vytvořeny pomocí verze 2024-02-03 rozšíření:



Nejprve ve VSC otevřeme složku se simulovanými soubory. V případě MHL2 jde o adresář: **C:\SPS\GHDL_MHL2**.



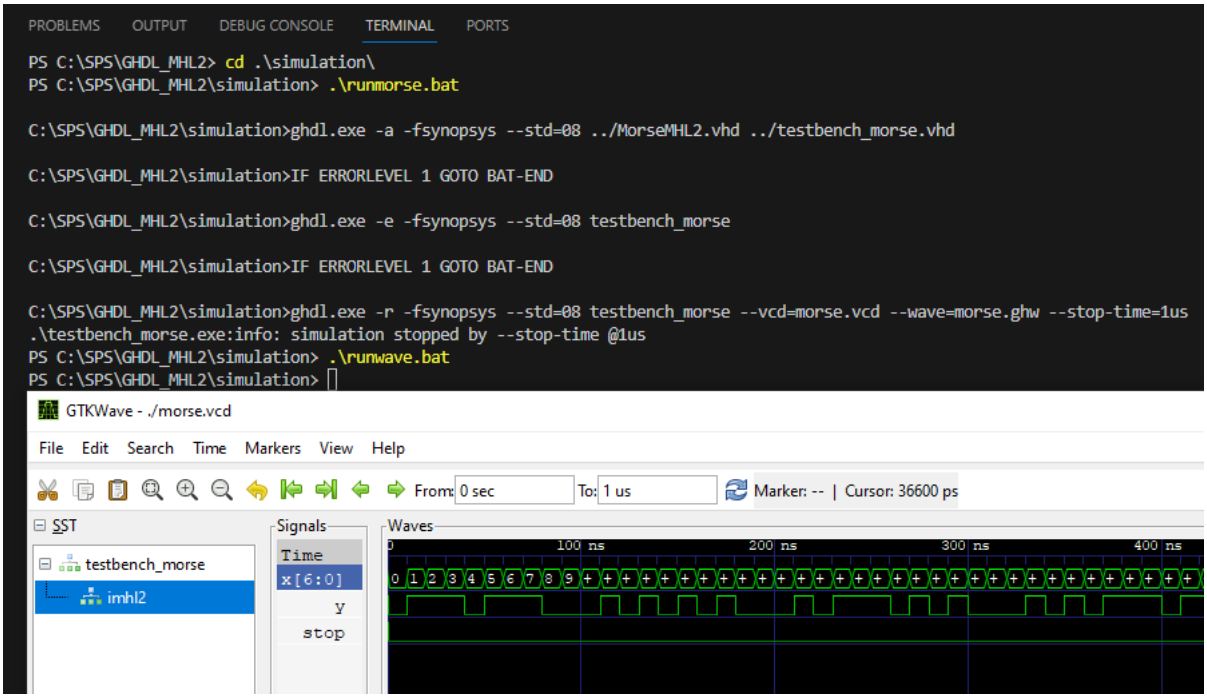
Po jeho otevření vytvoříme nový příkazový terminál:



Spustí se nám ve složce C:\SPS\GHDL_MHL2.

Napřed ji změníme na její podadresář **simulation**, kam jsme záměrně umístili soubory *runmorse.bat* a *runwave.bat,* aby tam zůstaly také všechny mezi-soubory GHDL kompilace a nezaneřádily nám hlavní složku.

Poté spustíme postupně dávkové soubory:



Poznámky:

- Okno VSC příkazového řádku akceptuje jako oddělovače složek jak znaky **/** tak **\.**
- Ve výchozím nastavení nenačítá Windows PowerShell za všech okolností soubory z aktuální složky. Zachovává zvyklosti systému Linux. Před všechny názvy souborů z aktuálního adresáře musíme tak přidat **./** nebo **.** prefixy.
- **Stisknutím klávesy [TAB]** se příkaz automaticky dokončí. Existuje-li více než jedna možnost, můžeme znovu stisknout klávesu [TAB] a zobrazit následující nabídky nebo pokračovat v psaní, až zůstane jediná odpovídající volba.

  Ve výše uvedeném příkladu jsme zapsali pouze:

cd ./	*po TAB se automaticky doplnilo na*	**cd .\simulation**
./r	*po stisknutí TAB se automaticky doplnilo na*	**.\runmorse.bat**
./r	*po dvou stisknutích TAB se objevilo*	**.\runwave.bat**

Simulace v GHDL se aktivuje rychleji než v interním simulátoru Quartusu a spustí se snáze než profesionální simulátor ModelSim, který sice umí více, ale žádá si tíž nastudování jeho prostředí dost odlišného od běžných programovacích nástrojů. Pomocí GHDL můžeme ladit i pokročilejší úlohy, například obvod, který úsporně vytvoří obrazové pozadí LCD řídícího panelu, což je následující zápočtový projekt našeho kurzu Logické systémy a procesory.

[Obrázek byl vytvořen užitím Corel 9 ClipArts]

~~~ Konec ~~~
~~~