



Projekt

Úlohy z OS Zephyr pro cvičení z OPS

Studijní program:

B0613A140005 – Informační technologie

Studijní obor:

B0613A140005AI – Aplikovaná informatika

Autor práce:

Dominik Šefr

Vedoucí práce:

Ing. Lenka Kosková Třísková, Ph.D.

Liberec 2024

Tento list nahradte
originálem zadání.

Prohlášení

Prohlašuji, že svůj projekt jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mého projektu a konzultantem.

Jsem si vědoma toho, že na můj projekt se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mého projektu pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li projekt nebo poskytnu-li licenci k jeho využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Beru na vědomí, že můj projekt bude zveřejněn Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

24. 5. 2024

Dominik Šefr

Úlohy z OS Zephyr pro cvičení z OPS

Abstrakt

Vytvořte kód testovacích úloh pro OS Zephyr určených k výuce OPS. Práce vychází z existujícího základu.

Klíčová slova: Zephyr, OPS, OS

OS Zephyr tasks for OPS exercises

Abstract

Create code for Zephyr OS test problems for teaching OPS. The work is based on an existing baseline.

Keywords: Zephyr, OPS, OS

Poděkování

Rád bych poděkoval všem, kteří přispěli ke vzniku tohoto díla.

Obsah

Seznam zkratek	8
1 Úvod	9
1.1 Cíl projektu	9
1.2 Struktura projektu	9
2 Zephyr	10
2.1 Historie	10
2.2 Co je RTOS?	10
2.3 Proč Zephyr místo Mbed?	10
3 Metodika	11
3.1 Vývojové prostředí	11
3.1.1 WSL	11
3.1.2 Visual Studio Code	11
3.1.3 Ubuntu	11
3.1.4 West	12
3.2 Nastavení vývojového prostředí	12
4 Implementace	13
4.1 Popis úloh	13
4.1.1 Hello	13
4.1.2 Config	13
4.1.3 Kernel	14
4.1.4 Threads	14
4.2 Zpětná vazba jednotlivých úloh	14
4.2.1 Hello	14
4.2.2 Config	14
4.2.3 Kernel	15
4.2.4 Threads	15
4.2.5 Zbylé úlohy	15
4.3 Řešení poskytnuté se zadáním	15
Závěr	16
Použitá literatura	17

Seznam zkratek

TUL	Technická univerzita v Liberci
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
OS	Operační systém
OPS	Operační systémy (předmět)
RTOS	Operační systém reálného času
WSL	Windows subsystem for Linux
VSC	Visual Studio Code

1 Úvod

1.1 Cíl projektu

Důvodem proč tato práce a její zadání vzniklo je ten, že se paní Ing. Lenka Kosková Třísková Ph.D. rozhodla přejít z výuky Mbed OS na Zephyr OS a potřebuje pro tyto hodiny vytvořit látku. Cílem tohoto projektu je důkladné ověření, zda navržené úlohy od paní doktorky Třískové pro její hodiny cvičení operačních systémů neobsahují chyby a jsou prakticky proveditelné. Projekt se zaměří na detailní analýzu každé úlohy, která bude zahrnovat jejich vyřešení a implementaci ve formě zdrojových kódů. Dále bude poskytována zpětná vazba k jednotlivým úlohám, identifikaci případných chyb a návrhy na vylepšení nebo úpravy. Výsledkem projektu budou tedy nejen samotné zdrojové kódy řešení úloh, ale i hodnocení a doporučení týkající se jejich kvality a realizovatelnosti.

1.2 Struktura projektu

Nejdříve se v práci zaměřím na technologie používané v projektu, abych poskytl přehled o nástrojích a prostředích, které budou využity při vývoji aplikací pro operační systém Zephyr. Tento přehled zahrnuje jak softwarové nástroje, tak vývojové prostředí, které umožňují efektivní a strukturovaný vývoj. Projekt je rozdělen do několika hlavních kapitol, které postupně pokrývají všechny důležité aspekty vývoje a implementace aplikací v Zephyru. Poté vysvětlím jaké úlohy jsem měl zpracovávat a co bylo jejich náplní a následně poskytnu zpětnou vazbu na zpracování těchto úloh.

2 Zephyr

Zephyr je malý operační systém reálného času vydán pod licencí Apache 2.0 pro připojená a vestavěná zařízení a zařízení s omezenými zdroji. Podporuje různé architektury, mezi které patří ARC*, ARM, Nios II, RISC-V, SPARC, Tensilica a x86. Obsahuje jádro a všechny potřebné komponenty, knihovny, ovladače atd. pro vývoj plnohodnotného aplikačního softwaru. [5]

2.1 Historie

Zephyr je založen na systému Virtuosu RTOS, který se v průběhu let přejmenoval na Rocket, od roku 2016 se k Wind River Systems přidala organizace Linux Foundation a vytvořili projekt pod názvem Zephyr. Rocket se poté začal používat jako komerční verze Zephyru, jelikož Wind River Systems požadovala platby za cloudové služby pro Rocket. Od doby, kdy se k Wind River Systems přidala organizace Linux Foundation začali Zephyr podporovat i velké společnosti jako je Intel, NXP, Synopsys, Bose a další.

2.2 Co je RTOS?

Operační systém reálného času je typ operačního systému. Jeho hlavním rozdílem oproti klasickým desktopovým operačním systémům (Windows, Linux) je garance splnění určité činnosti v časovém úseku. Desktopové operační systémy naopak mají možnost zastavit a počkat až se uvolní vlákno. [4]

2.3 Proč Zephyr místo Mbed?

Jeden z předních důvodů určitě bude možnost kromě ARM nasadit Zephyr na x86 architekturu a další, mbed je navržen pro ARM Cortex-M. Dalším důvodem bude to, že na Zephyr spolupracuje největší množství vývojářů a v roce 2022 měl také největší množství commitů z RTOS, z toho se dá předpokládat, že vývoj půjde rychle kupředu.

3 Metodika

3.1 Vývojové prostředí

Vývojové prostředí, které jsem použil bylo Visual Studio Code. Toto prostředí je velmi oblíbené mezi vývojáři díky své flexibilitě a široké škále dostupných rozšíření. Pro Zephyr RTOS je k dispozici několik rozšíření, které usnadňují práci, jako je například Zephyr Project Extension Pack, ale žádné z těchto rozšíření jsem v tomto projektu nepoužil.

Pro běh Zephyru jsem zvolil Windows Subsystem for Linux (WSL), konkrétně distribuci Ubuntu. WSL umožňuje běh Linuxového prostředí přímo v systému Windows, což usnadňuje práci s nástroji a knihovnami, které jsou primárně určeny pro Linux. Instalace a konfigurace Zephyru v Ubuntu byla relativně hladká a bez větších komplikací jsem mohl začít spouštět jednotlivé úlohy a testy.

Při pokusech o instalaci potřebných knihoven a nástrojů ve Windows Powershell jsem narazil na několik problémů. Powershell sice nabízí robustní prostředí pro plnění úkolů určené pro Linuxové prostředí, avšak v kontextu tohoto projektu se ukázalo jako méně vhodné. Konkrétní problémy a jejich řešení jsou podrobněji popsány v kapitole o řešení problémů. Nakonec jsem tedy zůstal u WSL a Ubuntu, což se ukázalo jako efektivní a stabilní řešení pro práci se Zephyrem.

3.1.1 WSL

Windows Subsystem for Linux je prostředí určené pro používání Linuxových nástrojů uvnitř operačního systému Windows. Hlavním důvodem je používání Bash a knihoven určených převážně pro Linux. [3]

3.1.2 Visual Studio Code

Visual Studio Code je editor zdrojového kódu od společnosti Microsoft, obsahuje možnost přidání velkého množství rozšíření pro různé jazyky. Ačkoli je Visual Studio Code od společnosti Microsoft, tak je "svobodným softwarem", což u vývojových softwarů od této společnosti není zvykem. [2]

3.1.3 Ubuntu

Ubuntu je distribuce operačního systému Linux založená na distribuci Debian GNU/Linux. Ubuntu je kompletně svobodná a otevřená distribuce. Mezi programátory

je velmi oblíbená, dlouhodobá podpora, výborná dokumentace. [1]

3.1.4 West

West je nástroj pro správu projektů a sestavení, navržený speciálně pro práci s operačním systémem Zephyr. Jeho hlavní funkcí je zjednodušení a automatizace úkolů spojených s vývojem softwaru pro embedded systémy. West je navržen tak, aby pomohl spravovat víceúlohové projekty, kde je potřeba koordinovat různé komponenty a závislosti. [5]

3.2 Nastavení vývojového prostředí

Do Visual Studio Code jsem si nainstaloval rozšíření pro jazyk C, ve kterém jsou úlohy napsané. Toto rozšíření poskytuje podporu pro syntaxi, zvýrazňování kódu, lintování a další užitečné funkce, které usnadňují práci s C. Díky tomuto rozšíření bylo možné psát a ladit kód efektivně a bez problémů. Jiné nastavení nebylo třeba, což značně usnadnilo celý proces nastavení vývojového prostředí.

4 Implementace

4.1 Popis úloh

Úloh je celkem 9 a poskytují procvičení základních dovedností začínajícího programátora v Zephyr RTOS. Úlohy jsou psané v anglickém jazyce.

4.1.1 Hello

První úlohou je úloha Hello, tato úloha provází prvním setkáním se Zephyr RTOS. První krok nás seznámí s instalací přes dokumentaci oficiálních vývojářů, kde jsou popsány instrukce pro různé operační systémy (Ubuntu, macOS, Windows). Doporučuje se použití virtuálního prostředí pro Python. Druhý krok je instalace Zephyru a Python dependencies, zde je popsán postup instalace virtuálního prostředí, instalace knihovny West a následná instalace operačního systému Zephyr. Třetím krokem je instalace vývojových nástrojů Zephyr SDK pro vývoj aplikací, zde se nainstaluje poslední kus potřebných souborů. Jelikož budeme používat desky `qemu_x86` a `qemu_m0` pro emulaci, tak nepokračujeme v následování oficiální dokumentace. Druhá část úlohy je spuštění aplikace Hello. Zde si máme přečíst dokumentaci ohledně vytváření, konfigurace a sestavení aplikací v Zephyr RTOS. Obsahuje detaily o konfiguraci pomocí Kconfig, použití devicetree overlay, nastavení CMakeLists.txt a důležité proměnné build systému. Zbývá už jen spuštění aplikace vytvořené pro test Zephyr emulátoru od paní Ing. Lenky Koskové Trískové Ph.D. Zbytek úlohy je věnován řešením problémů se spouštěním.

4.1.2 Config

Druhou úlohou je úloha Config, která se zaměřuje na nastavení konfiguračních souborů aplikace. Tato úloha zahrnuje definování hodnot konstant, nastavení zpoždění spuštění a konfiguraci úvodních zpráv systému. Konfigurace systému v Zephyru se provádí pomocí konfiguračních souborů, které používají syntaxi KConfig. Hlavní konfigurační soubor projektu je `prj.conf`, který může být rozdělen do více menších souborů pro lepší organizaci. V konfiguračních souborech můžete definovat různé konstanty, které ovlivňují chování systému. Například můžete nastavit zpoždění spuštění pomocí `CONFIG_BOOT_DELAY` a konfigurovat úvodní zprávu pomocí `CONFIG_BOOT_BANNER` a `CONFIG_BOOT_BANNER_STRING`. CMakeLists.txt je soubor používaný nástrojem CMake k definování procesu sestavení (build procesu) softwaro-

vého projektu. Tento soubor obsahuje instrukce a nastavení, které CMake potřebuje k vytvoření build systému specifického pro danou platformu, například Makefile na Unixových systémech nebo projektové soubory pro různé verze Visual Studia na Windows. Druhá úloha je zaměřena na menuconfig, guiconfig a KConfig. Menu a guiconfig jsou konfigurační způsoby více přizpůsobené uživateli, tím pádem není potřeba učit se syntaxi konfiguračních souborů. Poslední úloha je spíše dodatečná informace o tom, že několik proměnných prostředí také ovlivňují sestavení Zephyr.

4.1.3 Kernel

Úloha Kernel pojednává o jádru operačního systému a hlavně binárních souborech a jejich velikosti. První úloha je spíše teoretická, co jsou binární soubory, jak jsou velké atd. Druhá úloha je pozorovací, či testovací. Pozoruje se tu jak se mění jádro systému v závislosti na konfiguraci. Zda se zvětšují, či zmenšují binární soubory. Třetí úloha je už programátorská a vyžaduje nastudování dokumentace o časovačích, zde se požaduje napsat dvě verze kódu, jedna je program, který bude vypisovat frázi "Hello world" periodicky. Druhá verze je program, který vypisuje zprávu také periodicky, ale čas bude nastaven v konfiguraci aplikace.

4.1.4 Threads

Úloha Threads se věnuje vláknům. V této úloze se student naučí jak pracovat s vlákny v RTOS Zephyr. Úloha začíná jednoduchým spouštěním vláken, přejde do paralelního spouštění více vláken. Později se v úloze pojednává o komunikaci mezi vlákny pomocí trubek. V druhé části úlohy se studenti učí jak pracovat s uživatelskými vlákny, které jsou možné používat jen na systémech se správou a ochranou paměti.

4.2 Zpětná vazba jednotlivých úloh

4.2.1 Hello

Úloha Hello je velice jednoduchá na vyřešení, stačí se řídit dokumentací poskytnutou v zadání. Úloha je bezproblémová až na poslední část, kde je špatně vložená direktiva preprocesoru pro hlavičkový soubor kernel.h, je tedy potřeba změnit tuto direktivu z `#include <kernel.h>` na `#include <zephyr/kernel.h>`. Zbytek úlohy byl absolutně bezproblémový

4.2.2 Config

Úloha Config je spíše teoretická, než praktická. Poskytnutá dokumentace je přehledná a není problém celou úlohu mít hotovou za pár minut. Do `prj.conf` vepíšeme vlastní hodnoty a následně spustíme aplikaci, toto budeme opakovat než se dopracujeme k odpovědím na zadané otázky. Nastavení pomocí menuconfig, či guiconfig

není problémem a může se pokračovat dále. Zbytek úlohy je opět následování dokumentace a pokynů.

4.2.3 Kernel

Úloha Kernel už začíná s kódováním, není problém napsat lehký časovač opakující hlášku po časové prodlevě. Hlavní smyslem je zjistit jak se chovají binární soubory jádra při různých nastaveních konfiguračních souborů. Opět je cvičení plně funkční, takže splnitelné za 30 minut času.

4.2.4 Threads

Toto je už komplexně provedená úloha, zadání je srozumitelné. První část nebyl problém splnit za hodinu. Problém nastává u druhé části úlohy, kde se požaduje spustit kód z úlohy Hello, při které by měl zephyr vrátit **ASSERT** zprávu, Zephyr mi vrátil takovou zprávu, ale kvůli jinému důvodu, hodilo by se přidat příklad, či screenshot jak taková zpráva vypadá.

4.2.5 Zbylé úlohy

Zbylých 5 úloh nebylo možné splnit, jelikož nejsou hotové. Úloha Scheduling má sice úvod, ale není v ní žádný zadaný úkol.

4.3 Řešení poskytnuté se zadáním

Všechny řešení v úlohách poskytnuté se zadáním jsou plně funkční a přesně reprezentují požadované kódy.

Závěr

Rád bych poděkoval především paní Ing. Lence Koskové Třískové Ph.D. za poskytnutí možnosti spolupracovat na jejím projektu pro své hodiny OPS. Myslím si, že zvolením RTOS Zephyr neudělala chybu, ba naopak je to skvělý nástroj pro vývoj na RTOS zařízeních. Ačkoli mne mrzí, že zbylé úlohy nejsou dodělané, tak jsem si práci jako tester na tomto projektu užil a naučil se mnoho nových věcí.

Použitá literatura

- [1] CONTRIBUTORS, Wikipedia. *Ubuntu*. 2023. Dostupné také z: <https://cs.wikipedia.org/wiki/Ubuntu>. Accessed: 2024-05-22.
- [2] CONTRIBUTORS, Wikipedia. *Visual Studio Code*. 2023. Dostupné také z: https://cs.wikipedia.org/wiki/Visual_Studio_Code. Accessed: 2024-05-22.
- [3] CONTRIBUTORS, Wikipedia. *Windows Subsystem for Linux*. 2023. Dostupné také z: https://cs.wikipedia.org/wiki/Windows_Subsystem_for_Linux. Accessed: 2024-05-22.
- [4] CONTRIBUTORS, Wikipedia. *Real-time operating system*. 2024. Dostupné také z: https://en.wikipedia.org/wiki/Real-time_operating_system. Accessed: 2024-05-22.
- [5] CONTRIBUTORS, Wikipedia. *Zephyr (operating system)*. 2024. Dostupné také z: [https://en.wikipedia.org/wiki/Zephyr_\(operating_system\)](https://en.wikipedia.org/wiki/Zephyr_(operating_system)). Accessed: 2024-05-22.

A Přílohy

- Text práce