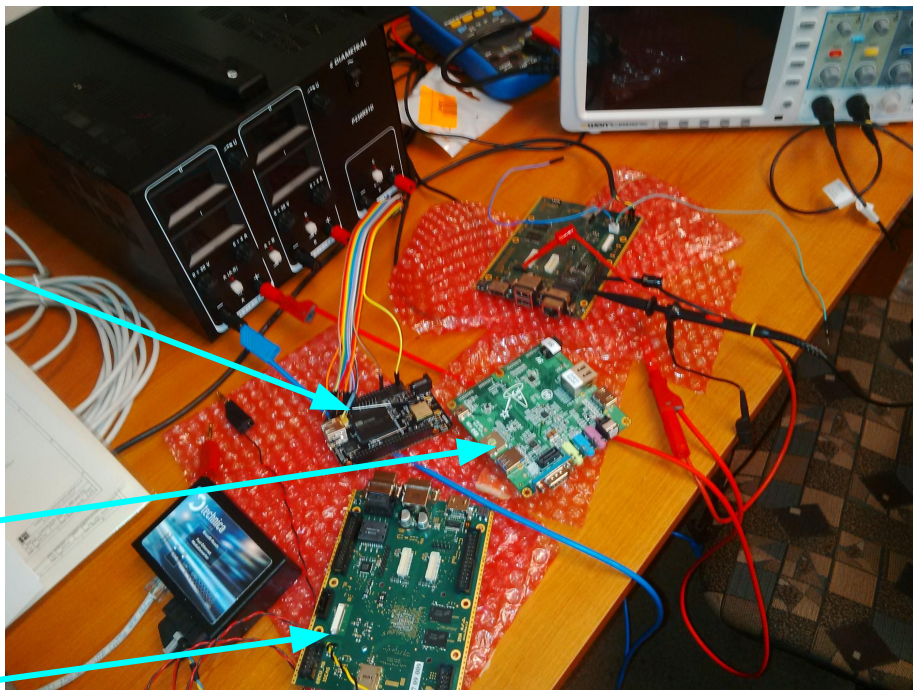


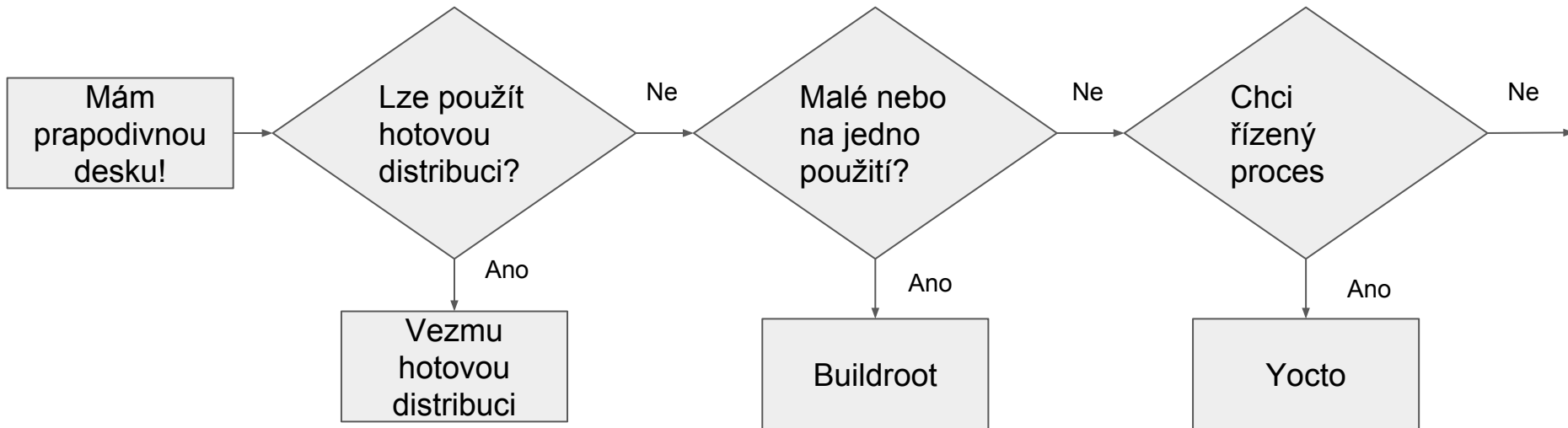
Linux na prapodivných deskách

LinuxDays 2017, Lenka Kosková Třísková
Technická univerzita v Liberci

Prapodivné desky?

- Se systémem od výrobce
(BeagleBoard black)
- S podporou pro vývoj vlastní distribuce
(WandBoard)
- Zákaznická deska bez podpory,
navíc s nestandardním HW





Prapodivné desky?

- *Single board computer*
 - Celý systém je na jedné desce plošných spojů.
 - Typicky celá řada pinů (sběrnice, čítače, analogové vstupy/výstupy)
 - Síť, wifi, USB, nějaké video (od VGA po několikanásobný full HD)
 - Procesor typicky “nějaký ARM”
 - Úložiště SD karta, někdy konektivita pro disky
- *System on chip - “nějaký ARM a něco”*
 - V jednom čipu typicky dohromady GPU, RAM, řadič USB a další
 - Často zajímavé periferie (AD/DA, CAN...)
 - Typicky nějaké jádro ARM plus výrobcem vyvíjené prvky
 - Samsung, Broadcom, Qualcomm, NXP, Nvidia, Renesas...

“Nějakej ARM”

- Cortex A = “Applications” - pro zákaznické systémy s plným OS (“počítače”)
 - Tablety, mobily, servery, robotika...
 - Vysoký výkon, 64/32bit
- Cortex R = “Real time” - embedded, signal processing
 - Automotive, kamery, úložiště...
 - Speciální péče věnovaná přerušením, obsluze paměti...
 - DSP support - bloky pro filtry, transformace...
- Cortex M = “Microcontroller” - pro MCU a SoC
 - Nízké taktovací frekvence, malá spotřeba
 - DSP support - bloky pro filtry, transformace, řízení motorů, statistiku

Embedded Linux

“Linux je free, nějak si ho přeložím.”

=> Potřebujeme:

- Překladač pro cílový HW
- Knihovny a podporu HW specialit
- Zavaděč operačního systému spustitelný na cílovém HW
- Další aplikace a funkce

Kdy to má cenu?

O Linuxu se vyplatí přemýšlet, když:

- Mám cca 64 MB RAM a více.
- Mám procesor “asijakopentium” tak na 200 MHz.
- Mám cca 100 MB na souborový systém.
- Mám dost energie.

Alternativy?

- RTOS, FreeRTOS
- TinyOS
- RIOT
- mBed

EmbeddedLinux = TBKR

Při vývoji musíme zajistit/vybrat/mít “TBKR”:

1. Nástroje pro překlad ze zdrojového kódu a zdrojové knihovny - **Toolchain**
2. Vhodný zavaděč - **Bootloader**
3. Sestavené přeložené jádro - **Kernel**
4. Souborový systém s připravenými knihovnami a programy - **Root Filesystem**

Toolchain

“Sada nástrojů, jež zajišťují překlad zdrojového kódu do formy spustitelné na cílovém zařízení.” (Mastering embedded Linux Programming)

Zahrnuje:

- překladač,
- linker,
- běhové knihovny (runtime libraries).

Co se bude překládat?

- zavaděč,
- jádro,
- a všechny doplňkové nástroje (shell, utility, ...)

Standardní a nestandardní toolchain

- **GNU (“normální Linux”):**
 - Binutils + gcc + glibc
- **Nativní:**
 - Zvolený toolchain lze spustit přímo v cílovém systému, překládáme na cílové architektuře.
- **Cross:**
 - Křížový překlad - pracujeme např. na rozumně rychlém PC, s architekturou odvozenou od x86, překládáme pro ARM.
- **Kde ho vzít?**
 - U výrobce (SoC? ARM?).
 - Sestavit samostatně (crosstool-ng).
 - Podpůrné systémy pro sestavování distribucí.

Je libo libc?

- “C library” - knihovny pro systémová volání a základní funkce OS
- V “normálním” Linuxu glibc
- V “embedded” mnoho verzí, protože můžeme mít problém s:
 - licence - nechceme vždy GPL (musl-libc),
 - velikost - máme málo paměti (uclibc),
 - architektura - výrobce má něco svého (avr-libc)
- Volba knihoven je zásadní, svazuje nás s dalšími nástroji, nelze měnit zpětně.

Zavaděč

1. Start kódu z nějaké paměti typu ROM
2. Hledá se kód ke spuštění v nějaké dostupné paměti nebo na síti nebo přes UART - toto určí výrobce HW.
3. Zavádí se OS - cílem je kopírovat do paměti jádro a nastartovat init.
 - **U-Boot**
 - **Barebox**

GRUB se nepoužívá (není pro ARM, trable s GNU v3 a secure boot).

- Zavaděč se nahrává na zařízení dle pokynů výrobce (typicky sektor na paměťové kartě, někdy vestavěná karta na desce).

Jádro

“Nastavit síť v našem Linuxu zvládne každá mamina od plotny, pokud ovšem ví, co je to IP vrstva.” (Prezentace nějaké distribuce na SLT 1998)

- Teoreticky je možné stáhnout si nejnovější zdrojáky z kernel.org a pustit se do práce... pokud ovšem víte, co je to IP vrstva.
- Prakticky:
 - Výrobce desky zveřejňuje aktualizované a testované jádro
 - Výrobce SoC zveřejňuje...
 - Výrobce architektury...
 - Yocto, Linaro, Buildroot...
- Nejde jen o vlastní jádro, ale o moduly a jejich výběr, sestavení device tree a také o ovladače případného speciálního HW.

Kromě jádra...

- Init
- Shell & Utilities
- Služby a démoni
- Knihovny
- Konfigurační soubory
- /dev, /proc, /sys
- Moduly jádra

A co ještě?

- Nastavit init
- Určit, co se spouští a v jakém pořadí
- Definice nastavení zařízení
- Konfigurace připojovaných souborových systémů
- Nastavení sítě
- Vytvoření uživatelů, nastavení oprávnění

Spousta dílčí práce, musí se dokumentovat, musí se hlídat verze nástrojů, problém s aktualizacemi... => Chybí tomu systém! => **Build systems!**

Build system

- Automatizace postupu TBKR (toolchain-bootloader-kernel-rootfs)
 1. Obstará zdrojové kódy dle zvolené konfigurace a HW, zajistí záplaty a opravy
 2. Postup sestavení systému probíhá podle sepsaného scénáře (= dokumentováno!)
 3. Automaticky vytváří jednotlivé komponenty
 4. Sestaví souborový systém
 5. Připraví image a všechny doprovodné soubory (bootloader nebo nějaké skripty)

Jaké to jsou?

- Yocto

- Zpřístupňuje nástroje pro grafické rozhraní systému (X11, GTK+, Qt, Clutter, SDL a další).
- Jádro je kompatibilní s OpenEmbedded Core.
- Součástí projektu je i podpora různých architektur pro emulátor QEMU (včetně jádra ARM 9).
- Vyvíjený systém je možné rozšiřovat a doplňovat další funkce.
- Pro zařízení s omezenou velikostí obrazovky nabízí Yocto *Sato reference user interface*, grafické uživatelské rozhraní vystavěné nad GTK+.

- BuildRoot

- Vytvářeno s cílem mít minimální image, minimální možná konfigurace, minimální závislosti
- Ve srovnání s výstupem Yocto složitě následné úpravy

Yocto

- Distribuce ve vrstvách, základem jádro a systémové knihovny
- Výrobci HW či jiní vývojáři přidávají vrstvy
- Pro novou desku se vytváří “Board Support Package”
- Konfigurace překladač - BitBake & Recipes

Reálně: Deska WandBoard, podpora od výrobce již existuje - cca 1 hodina pro nastavení, cca 6 hodin pro překladač, image včetně SATO - 372 MB

Zdroj obrázku - dokumentace k Yocto

