FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Systém pro hromadnou správu programového vybavení robotů
Student:	Petr Kolář
Vedoucí:	Ing. Miroslav Skrbek, Ph.D.
Studijní program:	Informatika
Studijní obor:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Navrhněte a realizujte systém pro hromadnou správu programového vybavení robotů. Systém musí poskytovat paralelní nahrávání firmwaru do skupiny robotů přes USB z víceportového hubu. Musí zajistit jednoznačnou identifikaci instancí robotů a selektivní nahrávání různých programů do podskupin robotů. Systém musí také umět spravovat roboty založené na operačním systému Linux. Pro tyto roboty je požadována jednoznačná identifikace instancí, autentizace prostřednictvím certifikátů, rozdílové aktualizace kořenového souborového systému, instalace aplikací a možnost přenášení souborů mezi počítačem spravujícím roboty a připojenými roboty. Systém navrhněte pro operační systém Linux. Rozhraní programového vybavení pro správu robotů navrhněte tak, aby bylo možné snadné napojení na webové rozhraní. Rozsah práce upřesněte po dohodě s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 12. února 2018

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Systém pro hromadnou správu programového vybavení robotů

Petr Kolář

Katedra číslicového návrhu
Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.

14. května 2018

Poděkování

Chtěl bych poděkovat Ing. Miroslavu Skrbkovi, Ph.D. za vedení této práce a praktickou pomoc při vytváření systému, zejména za pomoc s robotem NAO. Dále bych také rád poděkoval své rodině a přítelkyni za podporu a pomoc při studiu.

Bakalářská práce byla spolufinancována Evropskou unií z Operačního programu Výzkum, vývoj a vzdělávání.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla. Souhlasím s tím, aby Dílo bylo veřejně šířeno pod některou z licencí Creative Commons 4.0 ve variantě BY a BY-SA nebo GPLv3.

V Praze dne 14. května 2018 .....

Odkaz na tuto práci

Kolář, Petr. *Systém pro hromadnou správu programového vybavení robotů.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato práce se zabývá návrhem a vytvořením systému pro správu robotů NAO, linuxových zařízení a vývojových desek Arduino. Navržený systém je schopný spouštět aplikace na zařízeních, vytvářet zálohy zařízení, restart zařízení, detekovat online zařízení a další funkce. Literární rešerše se zabývá analýzou možností identifikace jednotlivých zařízení, otázkou bezpečného spojení se zařízením a efektivního přenosu souborů mezi zařízením a uživatelem. Umožňuje také napojení na webový portál, který obsluhuje systém. Systém obsahuje možnost přidělení jména Arduino zařízení, tato možnost není běžná u podobných systémů. Příloha práce obsahuje uživatelskou příručku systému.

Klíčová slova systém správy robotů, robot NAO, správa vývojových desek Arduino, laboratoř robotů ČVUT, Linux, záloha robotických zařízení, identifikace Arduino zařízení, SSH, udev

Abstract

This thesis deals with design and develop system for the managing NAO robots, linux devices and develob Arduino boards. The designed system is able to run the application, create backup of the devices, restart devices, and provide more functions. The literature search is about analyzing possibilities of identification various devices, question of secure connection with device and effective transfer of files between device and user. It enables to join with the web portal, which handles to the system. The system allows to give a name to Arduino device, this feature is not common in similar systems. Attachment of thesis contains user reference book for the system.

Keywords Robot management system, NAO robot, Arduino boards management, CTU robot laboratory, Linux, backup of robotic devices, identification of Arduino boards, SSH, udev

Obsah

Seznam obrázků

Úvod

Ve světe se neustále zvyšuje počet inteligentních zařízení a s jejich rostoucím počtem rostou také nároky na jejich správu. To vede ke vzniku systémů, které dokážou takováto zařízení spravovat. Specifickou aplikací těchto systémů jsou vývojové a výukové laboratoře. Po takovýchto systémech je požadováno, aby dokázaly zajistit snadnou výměnu softwaru v zařízeních, jednoduché spouštění aplikací na zařízeních a zálohu dat. To vše při vysokém stupni zabezpečení.

Rozšíření výuky robotizace na ČVUT FIT vede k nutnosti vytvoření takového systému pro správu robotů a Arduino zařízeni. Tato zařízení se budou využívat k výuce v nových laboratořích inteligentních vestavných systémů. V budoucnu bude nakoupeno několik nových robotů založených na systému Linux a také několik různých Arduino zařízení. V této práci se zabývám připojováním těchto robotů a Arduino zařízení do systému pro jejich správu. Tento systém bude využíván studenty a vyučujícími FIT. Systém výrazně ulehčí hodnocení studentů a zjednoduší nahrávání softwaru do robotů a obecně práci s těmito zařízeními. Téma jsem si zvolil z důvodu absence podobného systému na fakultě a také pro svůj zájem o tato zařízení.

V práci se zabývám analýzou možností identifikace jednotlivých zařízení, dále se zabývám metodami připojení jednotlivých zařízení do systému, přístupem systému k databázi a bezpečným přenosem dat mezi zařízeními a serverem. Tato bakalářská práce je součástí projektu rozsáhlého informačního systému laboratoře vestavných systémů, který umožní studentům přistupovat k zařízením přes webový portál. Přímo navazuje na práci studenta Dana Zatloukala, který se zabývá vytvořením databáze pro tento systém a implementací jádra webového systému, a Erika Zatloukala, který vytváří vzhled toho webu a uživatelské prostředí.

Kapitola 1

Cíl práce

Cílem rešeršní části práce je nastudování potřebných technologií k vytvoření bezpečného a efektivního systému s jednoznačnou identifikací zařízení, zejména protokolu SSH pro bezpečné propojení jednotlivých robotů se serverem a programu rsync pro efektivní přenos dat a vytváření záloh zařízení. Jedním z dílčích cílů je také prozkoumat možnosti jednoznačné identifikace zařízení typu Arduino a linuxových zařízení a rozpoznání připojení takového zařízení v systému.

Cílem praktické části je na základě zjištěných skutečností vybrat vhodná řešení pro systém a na jejich základě systém realizovat. Cílem je, aby realizovaný systém splňoval požadavky na jednoznačnou identifikaci instancí, jednoduché přenášení souborů mezi počítačem a serverem, podporoval instalaci aplikací, spouštění aplikací a zálohu zařízení. Součástí požadavků je vytvořit rozhraní systému, které bude snadno napojitelné na vznikající webové rozhraní.

Kapitola 2

Analýza a návrh

2.1 Současný stav

V současnosti při výuce v laboratořích katedry číslicového návrhu neexistuje systém pro hromadnou správu linuxových zařízení nebo zařízení Arduino. Studenti nahrávají software do konkrétních zařízení, ke kterým jsou přímo připojení.

V případě robota NAO se nahrávání softwaru provádí pomocí aplikace Choregraphe nebo se přímo nahrává do robota pomocí SSH připojení. Při nahrávání pomocí Choregraphe může student vybrat robota z nabídky podle jeho jména nebo jeho IP adresy. Po připojení k robotovi již může vytvářet různé scénáře pro chování robota a pomocí tlačítka „Play“ je do něj nahrát. Pokud student chce nahrávat software do robota NAO přes SSH musí znát jeho IP adresu a heslo k uživatelskému účtu. Mnoho uživatelských účtů je náročné na správu. Nyní je k dispozici pouze jeden robot a není problém si pamatovat přístupové údaje a adresu robota. Příští rok však bude laboratoř vybavena novými roboty a současný přístup již bude nevhodný.

Při nahrávání do Arduino desek se používá Arduino IDE[2]. Studenti nahrávají do jednotlivých vývojových desek připojených přímo k počítači. Toto řešení je pro výuku vyhovující. Problém nastává ve specifických případech, kdy je třeba nahrávat software do více zařízení najednou nebo pokud je připojeno k počítači více vývojových desek. Po připojení jsou jednotlivé Arduino desky pojmenované pouze názvem portu a názvem typu zařízení. Identifikaci Arduino desky lze provést pomocí tlačítka „Získat informace o Desce“, které přečte deskriptory zařízení a zobrazí některé informace (VID, PID, název desky a případné sériové číslo desky). V případě desek stejné výrobní řady tedy zbývá rozlišení pouze pomocí sériového čísla, které není pro člověka vždy jednoduché rozlišit a navíc není sériové číslo vždy uvedeno. Arduino IDE nepodporuje nahrávání do skupiny desek a nahrávání do mnoha desek může být časově náročné.

Ostatní linuxová zařízení typu Raspberry Pi a podobné se používají in-

dividuálně, proto pro ně nejsou žádné specifické postupy pro použití v laboratořích číslicového návrhu.

2.2 Existující řešení správy robotů

Většina vývojových laboratoří používá systémy pro správu chytrých zařízení. Většinou se jedná o systémy vyvíjené na míru jednotlivým laboratořím, s jasně specifikovanými požadavky a optimalizované pro dané použití. Součástí těchto systémů jsou rozsáhlé monitorovací funkce, které slouží ke kontrole zařízení a k ladění nahraného softwaru.

Na druhé straně jsou univerzální systémy pro širokou veřejnost. Tyto systémy nabízejí široké využití ať už v domácnostech nebo ve školách. Většinou se jedná o cloudová řešení, která jsou přístupná zdarma, či za malý poplatek (kolem 1$ za měsíc). Tato řešení často poskytují také monitorovací funkce zařízení a editor kódu.

2.2.1 Arduino Create

Arduino Create[3] je cloudová aplikace od společnosti Arduino.cc. Nabízí možnost tvorby sketchů v online editoru, jejich import do cloudu a snadné vyhledávání jednotlivých sketchů. Při tvorbě projektů pomůže Library manager, který obsahuje nejnovější sadu knihoven bez nutnosti instalace. Do Library manageru je možné přidávat vlastní knihovny. Součásti aplikace je i sériový monitor, který umožnuje monitorovat komunikaci po sériové lince a posílat zprávy do zařízení. Instalace Arduino Creat Agentu umožní rozpoznávat připojené desky k počítači a nahrávat software do zařízení přímo z webového prohlížeče. Cloudový editor představuje ořezaný editor z desktopové verze (například neumí exportovat *.hex soubory nebo specifikovat programátor pro zařízení). Velkou výhodou cloudového řešení je napojení na Arduino Project Hub, který obsahuje mnoho návodů a inspirací pro projekty.

Součástí Arduino Creat je i Arduino Cloud pro správu nejen Arduino Cloud zařízení (jako je například ARDUINO MKR1000, nebo je možné takové zařízení vytvořit pomocí WIFI shield 101), ale i zařízení Raspberry Pi, UP2 Board, BeagleBone nebo i pro některá zařízení založená na ARM a Intel x86 procesorech (teoreticky i robot Nao, ale tuto variantu jsem netestoval). Zařízení (testováno na Raspberry Pi) je po zadání IP adresy a přístupových údajů přidáno do Device manageru. Zde je možno zařízení pojmenovat, a pokud je online, lze o něm získávat informace (např. volné místo v zařízení nebo informace o připojení k síti). Do kompatibilních zařízení je také možno nahrávat software, instalovat aplikace nebo spravovat repositář. Ze zařízení, která podporují Arduino Cloud, jde číst informace ze senzorů.

Jako nevýhodu spatřuji nízkou uživatelskou rozšiřitelnost tohoto systému o nové funkce a špatnou detekci online zařízení (bylo zaznamenáno při testování na Raspberry Pi). Naopak jako výhodu bych zmínil komponentu Cre-

ative Technologies in the Classroom[4], která slouží k výuce programování Arduino zařízení. Komponenta umožnuje snadnou komunikaci mezi učiteli a studenty.

2.2.2 Dataplicity

Dataplicity[5] je aplikace, která umožňuje vzdálený přístup k Raspberry Pi. Aplikace poskytuje zabezpečené spojení přes HTTPS k Dataplicity agentu, který běží na zařízení. Agent umožňuje přístup k zařízení pomocí vzdáleného shellu, který běží buď v aplikaci, nebo na webové stránce. Pomocí něj pak lze zařízení ovládat podobně jako přes SSH, ale bez nutnosti složité konfigurace a vytváření spojení. Další vlastností Dataplicity je možnost vytvořit tzv. „černou díru“[6], která umožňuje zobrazit webovou stránku, kterou lze vytvořít a vystavit na portu 80 na Raspberry Pi. Ta pak může zobrazovat různé informace o zařízení. Součástí aplikace je i možnost nadefinovat si různé akce, které pak pouhým stisknutím tlačítka vykonají uživatelem nadefinované příkazy. K tvorbě akcí je možno využít i různé progress bary, posuvníky nebo textová pole. V aplikaci lze také provést diagnostiku zařízení a opravit ho. Dále aplikace umožňuje přidání různých nástrojů, které jsou však zpoplatněny. Aplikace umožňuje připojení jednoho zařízení zdarma s omezenými funkcemi. Další zařízení a prémiové funkce, mezi které bohužel patří i možnost SCP přenosu souboru, jsou zpoplatněné. Dataplicity poskytuje jednoduché REST API pro správu zařízení a uživatelského účtu.

Hlavní výhodou Dataplicity je jistě vzdálený přístup k shell. Další výhodou může být definování vlastních akcí. Jako nevýhodu spatřuji složitý přenos souborů do zařízení a z něj, který je navíc vázán na zpoplatněnou verzi.

2.3 Identifikace zařízení

Pro rozpoznání zařízení, která budou připojena k systému, musí systém umět rozpoznat jednotlivá zařízení pomocí rozpoznávacího znaku. Tento znak je složitější nalézt především u velmi jednoduchých zařízení, která mají omezenou paměť a omezený přístup k ní.

Arduino zařízení poskytují několik možností identifikace, přičemž ne všechny jsou podporovány všemi deskami. Podle zdroje[7] jsou hlavními metodami:

- Sériové číslo FTDI převodníku sériové linky. To může být přečtené pomocí ovladače od FTDI[8], který poskytuje API pro správu zařízení. Nové Arduino desky již tento převodník nevyužívají.

- Speciální hardwarový čip (obrázek 2.1), který obsahuje unikátní identifikační údaje zapsaný většinou v EEPROM. Tento čip komunikuje se zařízením např. pomocí I^2C. Nevýhodou tohoto řešení je, že čip zabírá piny na desce.

Obrázek 2.1: Hardwarový ID čip[1].

- Zápis vlastního identifikačního čísla do paměti EEPROM. Tento přístup je nejjednodušší, stačí zapsat ID na vhodnou adresu paměti, ta však může být přepsána při běžném používání desky. Další nevýhodou je že některé zavaděče (např. Optiboot) nepodporují čtení EEPROM ze strany počítače.

- Sériové číslo jako usb device descriptor. Převodník sériové linky obsahuje device deskriptory, které popisují samotné zařízení. Některá Arduina kromě údajů potřebných pro popis daného typu zařízení (VendorID, ProductID) a zároveň důležitých pro nahrávání softwaru do zařízení, obsahují také sériové číslo zařízení. Zařízením, která tento deskriptor neobsahují a mají převodník sériové linky, může být tento deskriptor změnou firmwaru doplněn.

- Vložení sériového čísla do zavaděče. Tato metoda je nejuniverzálnější a podporuje i zařízení, která neobsahují vlastní sériový převodník. Zároveň ale patří k těm složitějším.

Identifikace zařízení robotů NAO a ostatních linuxových zařízení je velmi rozmanitá. Může být prováděna například:

- identifikací na základě sériového čísla komponenty (CPU, HDD, ...)

- MAC adresou síťového rozhraní (speciální, případ předešlé varianty)

- zápisem sériového čísla do konfiguračního souboru

- pomocí pevné IP adresy

Robot NAO navíc obsahuje dvě speciální sériová čísla a to sériové číslo jeho hlavy a sériové číslo těla. Bohužel tato čísla nejdou přečíst pomocí NAOqi api.

2.4 udev

Důležitou schopností systému je rozpoznání připojení zařízení. V případě Arduino zařízení je toho dosaženo pomocí manažeru udev (userspace `/dev`)[9]. Tento manažer spravuje bloková a textová zařízení v adresáři `/dev`. Manažer se skládá ze tří částí démonu udevdm, udev pravidel a nástroje udevadm pro správu udev. Démon naslouchá jádru systému a zachytává od něj zprávy (uevent). Tyto zprávy jsou přenášeny pomocí socketu Netlink. Na základě těchto zpráv pak spouští jednotlivé akce, které jsou dány udev pravidly. Informace o zařízení čte udev z `/sys`, kde jsou pomocí sysf (pseudo souborový systém) přenášeny z prostoru jádra do uživatelského prostoru.

Pomocí udev pravidel[10] si můžeme vytvořit vlastní pravidla pro zařízení. Pravidla pro udev jsou umístěna v `/etc/udev/rules.d` nebo v `/lib/udev/rules.d` (zde se nacházejí výchozí pravidla pro systém). Vhodnější je umísťovat pravidla do `/etc/udev/rules.d`. Pravidla umístěná v `/etc/udev/rule` mají vyšší prioritu. Priorita se řídí také podle názvu souboru pravidla. Formát názvu je „98-name_of_rule.rules“. Kde 98 je priorita pravidla (priorita pravidel je řízena lexikálně, čím nižší, tím se pravidlo vykoná dřív a může být přepsáno jiným), následuje název pravidla a přípona pro pravidla „.rules“. Samotná pravidla obsahují „klíčové hodnoty“, podle kterých se určuje, kdy se pravidlo spustí a také dané akce, které se provedou po spuštění. Klíčové hodnoty mohou být atributem zařízení (attr), systémovou proměnou (env), typem akce (remove/add), názvem subsystému (subsystem), názvem zařízení, které mu přiřadil systém (kernel) a další. Pokud uvede název klíčové hodnoty s koncovkou „s“, pak se vztahuje nejenom na dané zařízení, ale i na rodičovská zařízení (je vhodné využít tuto koncovku, protože málo kdy jsou veškeré informace dostupné přímo z koncového zařízení). Udev pravidla dále také obsahují různé proměnné např. proměnou pro výstup ze spuštěného programu(`%c`) nebo název zařízení přiřazeného systémem (`%k`). Akce mohou spouštět program (RUN, PROGRAM), načíst modulu, vytvořit symbolický název zařízení (SYMLINK), nastavovat práva přístupu k zařízení (CHMOD) a další.

Pro testování pravidel a jejich ladění lze využít správce udevadm[11]. Tento správce obsahuje mnoho nástrojů. Pro tvorbu udev pravidel se hodí nástroj info, který zobrazí atributy nebo proměnné zařízení. Pro zobrazení zpráv, které zasílá jádro, slouží nástroj monitor. Pro nastavení udev démona slouží nástroj control, který umí nastavit např. časové limity pro odezvu démona nebo znovu načíst změněná pravidla. Správce ještě obsahuje nástroje pro testování pravidel (test), sledování fronty zařízení (settle) a spouštění událostí (trigger).

2.5 Protokol SSH

Tato kapitola čerpá ze zdrojů [12] a [13]

Protokol SSH umožňuje bezpečné spojení s jednotlivými zařízeními. Protokol existuje ve dvou verzích SSH–1 a SSH–2. SSH–1 je zastaralou verzí, která je náchylná na útoky „Man in the midle“ a je nahrazena verzí SSH–2. Protokol využívá šifrované spojení mezi klientem a serverem. Pokud je na zařízení spuštěn SSH server, který ve výchozím stavu naslouchá na portu 22, je možné se k němu připojit pomocí klienta. Při zahájení komunikace se využívá asymetrické šifrování a po zaslání klíče klientem se začne používat symetrické šifrování. Po zahájení šifrované komunikace následuje autentizace klienta, která může být provedena několika způsoby, nejčastěji se využívá autentizace pomocí hesla a uživatelského jména nebo pomocí klíčů. Přenos dat probíhá na několika vrstvách[14]. První vrstvou je transportní vrstva, která se používá k počáteční výměně klíčů, nastavění parametrů přenosu a poté k přenosu šifrovaných dat. Zajišťuje integritu přenášených dat, jejich zašifrování a odšifrování a případnou kompresi. Další vrstvou je vrstva pro autentizaci uživatele, která předává serveru autentizační údaje o klientovi. Tato vrstva se využívá po ustanovení zabezpečeného spojení a využívá toto spojení. Poslední vrstvou je vrstva připojení. Po připojení jsou přenášeny transportní vrstvou různé SSH kanály (sloužící k obsluze různých terminálů nebo služeb). Tato vrstva slouží k otvírání nových kanálů, k jejich předávání jednotlivým službám a řízením jejich toku. Po autentizaci je pak možné na serveru spouštět různé příkazy a přistupovat k jeho souborů jako běžný uživatel. Server naopak nemůže nic spouštět na připojeném klientu. Pro snížení rizika útoku MITM umožňuje SSH podepsání serverových i klientských klíčů pomocí certifikátů.

2.5.1 Vytvoření spojení

Při vytváření spojení se nejprve klient připojí pomocí TCP k serveru na port 22. Následně si se serverem vymění informace důležité pro spojení a server se prokáže svým otiskem. Tento otisk je většinou odvozen od veřejného klíče serveru[15]. Po prvním přihlášení se uloží do souboru known_host. Při změně otisku se klient odmítne připojit k serveru, změna může indikovat útok MITM (může se však také jednat jen o změnu klíče serveru). Poté co se server a klient shodne na metodě symetrické šifrování (využívá šifry aes, 3des, twofish, ...) a na metodě zajištění integrity dat (hašovací algoritmy md5 a sha), klient vygeneruje náhodný klíč a dojde k výměně klíčů k symetrickému šifrování. Při výměně klíčů se používá Diffie–Hellmanův algoritmus (RSA v protokolu SSH–1). Poté co došlo k výměně klíčů, prokáže se server zasláním zprávy zašifrované symetrickou šifrou. Nyní bylo vytvořené zabezpečené spojení, které umožnuje bezpečnou autentizaci k serveru a předávání dat.

2.5.2 Metody autentizace

Nejběžnější metodou autentizace je autentizace pomocí hesla a uživatelského jména. Při této autentizaci zašle klient autentizační údaje přes zabezpečené spojení. Server ověří údaje, a pokud souhlasí, povolí přístup uživateli k serveru. Pokud ne, může server zažádat o znovu zadání autentizačních údajů. Autentizace pomocí hesla je nevyhovující k automatickému připojování, protože příkazu ssh nezle předat hesla přes standardní vstup ani pomocí přepínače.

Připojení pomocí SSH klíčů je bezpečnější metodou. Nedochází při ní k přenosu hesla. Ověření probíhá pomocí spárované dvojice klíčů (veřejný a soukromý), které jsou spárovány pomocí šifrovacího algoritmu rsa (další možnosti jsou dsa, ecdsa a ed25519). Nejprve je vytvořena pomocí příkazu ssh-keygen dvojce klíčů. Při vytváření můžeme zvolit pro soukromý klíč ochranu heslovou frází. Heslová fráze musí být zadána před použitím klíče a chrání tak klíč proti odcizení. Heslová fráze může být uložena v paměti pomocí agentu (např. keychain) a není ji tedy nutné zadávat při vytváření spojení (fráze je držena v paměti tedy po restartu zařízení je ztracena). Veřejný klíč může být zaslán serveru přes zabezpečené spojení, kdy byl klient prvně autentizován pomocí hesla, nebo může být přenesen i jiným způsobem. K přenosu klíčů může být také použit program ssh-copy-id. Klíče jsou na serveru uloženy v souboru authorized_keys. Samotná autentifikace probíhá tak, že server vygeneruje náhodný řetězec, který pošle klientovi. Následně klient klíč zašifruje svým soukromým klíčem a pošle zašifrovaná data zpátky serveru. Pokud se podaří serveru zaslaná data rozšifrovat, prokáže tak, že klient zná soukromý klíč, a povolí mu přístup.

Rozšíření autentizace pomocí hesla představuje keyboard-interactive metoda[16], která umožnuje modifikaci žádosti od serveru. Umožnuje například překlad požadavku na heslo do lokálního jazyka, zadání několika hesel nebo určité nastavení spojení.

Autentizace pomocí GSSAPI[17] (Generic Security Service Application Programming Interface) je metoda jak si vytvořit vlastní mechanismus přihlášení pomocí GSSAPI. Tato metoda je často spojována s protokolem Kerberos.

2.6 rsync

Tato kapitola čerpá ze zdroje [18]
Rsync je nástroj pro pokročilou synchronizaci a přenos souborů. K přenosu dat přes síť se může využít zabezpečený kanál, který vytváří SSH nebo je možné vytvořit samostatný rsync server. Hlavní výhodou rsync je ušetření datového toku tím, že při kopírování souborových struktur přesouvá pouze soubory, které se liší od souborů, jež jsou v cílové struktuře. K tomuto dochází typicky při zálohování systémů. Při tomto použití dokáže rsync významně ušetřit objem přenášených dat díky použití delta algoritmu. Tento algoritmus je zjednodušeně založen na rozdělení souborů na části, vytvoření kontrolních součtů

těchto částí a porovnávání jejich kontrolních součtů. Rsync při kopírování umožňuje zapnout pokročilou kompresi, čímž ještě víc sníží datový tok. Obsahuje také mnoho nástrojů vhodných pro zálohu dat a efektivní kopírování, například pro vytváření záplat, obnovení z bodů zálohy, zachovávání práv, časů modifikace a mnoho dalších.

2.7 Analýza požadavků a nástin řešení

Na základě požadavků ze zadání práce a požadavků, které vzešly z konzultací s vedoucím práce, byla vypracována základní funkcionalita systému, kterou systém zprostředkovává webovému portálu. Tyto požadavky jsou zprostředkovávány pomocí skriptů, které jsou uloženy na serveru. Tyto požadavky jsou:

- **Snadné napojení na webové rozhraní**

  Skripty jsou psány pro bash interpret. Tento interpret je na serveru umístěn v `/usr/local/bin/`. Webový portál, který využívá PHP, může spouštět tyto skripty, nastavovat je pomocí přepínačů a předávat jim pomocí přepínačů uživatelův vstup. Tyto přepínače vyžadují mezeru mezi přepínačem a hodnotou. Skripty obsahují omezenou kontrolu vstupů a základní prvky zabezpečení. Přesto některé prvky zabezpečení musí být zprostředkovány ze strany jádra portálu (např. při přenášení souborů by si uživatel mohl vybrat pro přenesení jakýkoliv soubor, ke kterému má skript přístup).

- **Implementace systému pro systém Linux**

  Na centrálním serveru běží operační systém FreeBSD ve verzi 11.1. Na připojených zařízení se předpokládá velká rozmanitost linuxových distribucí. Na robotu NAO běží linuxová distribuce Gentoo a systém využívá funkce NAOqi. Některé funkce systému (udev) mohou být závislé na dané linuxové distribuci a je nutné otestovat funkce systému při připojení nového zařízení a případně upravit vlastnosti systému.

- **Jednoznačná identifikace instancí**

  Pro identifikaci linuxových zařízení se používá jejich jméno, které je uvedeno v konfiguračním souboru zařízení, a IP adresa, která je nastavena u těchto zařízení napevno. Další možností identifikace těchto zařízení představuje MAC adresa zařízení. Tato možnost identifikace v systému je a systém si ukládá mac adresu zařízení, ta ale není využita k identifikaci. Pro identifikaci Arduino zařízení byla vybrána identifikace pomocí sériového čísla, které je uvedené v usb device deskriptoru. Tato metoda představuje jednoduchou možnost spojení identifikace zařízení a identifikace připojení zařízení pomocí udev. U zařízení, které tuto metodu nepodporují (např. arduino Esplora, která se nyní používá k výuce

předmětu BI–ARD), bude nutná úprava firmwaru, nebo implementace nové metody identifikace zařízení.

- **Autentizace prostřednictvím certifikátu**

  Pro autentizaci zařízení při spojení přes SSH byla vybrána metoda zabezpečení pomocí klíčů. Tato metoda přináší výhody především při automatickém zpracování a zároveň je jednoduchá na implementaci. Možnost zabezpečení klíčů za použití certifikátů byla s vedoucím práce konzultována a bylo shledáno, že v této fázi systému je její aplikace komplikovaná a její případná implementace byla odložena. Toto bylo učiněno také z toho důvodu, že zařízení nemají přístup k systému a případný útočník by mohl odcizit pouze data mířící na zařízení.

- **Paralelní nahrávání firmwaru**

  Pro nahrávání firmwaru do zařízení Arduino se využívá program avrdude. Tento program umožňuje náhrávat hex kód do zařízení s mikrokontrolérem z rodiny avr. Tento typ mikroprocesoru obsahuje většina Arduin (výjimkou může být např. Arduino Galileo s procesorem x86). Hex kód je třeba prvně vygenerovat např. pomocí Arduino IDE. Ten je přeložen z C kódu, který je vytvořen v IDE pomocí cross kompilátoru. Hex kód je pak přenesen na zařízení, ke kterému jsou připojena Arduina, a následně je paralelně spuštěn avrdude. Ten nahraje kód do flash paměti požadovaných Arduin.

- **Rozdílové aktualizace, přenášení souborů mezi počítačem a zařízením**

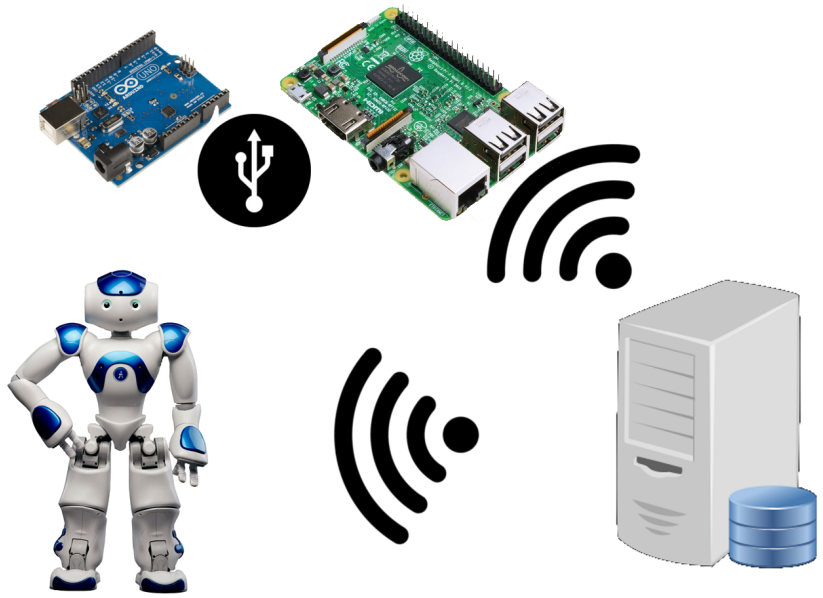
  Pro aktualizace kořenového souborového systému je využit program rsync, pomocí kterého systém vytváří zálohu zařízení nebo posílá data ze zařízení. Systém umožňuje vytvoření bodu obnovy robota a následné obnovení z této zálohy. Dále systém umožňuje vytvořit záplatu robota na základě rozdílů mezi provedenou zálohou a systémem uvedeným do nového stavu.

- **Instalace aplikací na robota NAO**

  Systém umožňuje instalovat aplikace, které jsou zabaleny ve formátu tar.gz2. Po rozbalení je aplikace nainstalována do kořenového adresáře zařízení. Tuto funkci podporují pouze zařízení s operačním systémem Gentoo (tedy i robot NAO).

- **Kontrola stavu zařízení**

  Pro webový portál je důležité, aby věděl, zdali je zařízení ve stavu online nebo offline. Toho je v systému dosaženo pomocí dvou mechanismů. Prvním mechanismem je jednoduchý sken IP adres, který pomocí příkazu ping zjišťuje stav zařízení a následně zapisuje stav zařízení



Obrázek 2.2: Architektura systému.

do databáze. Druhá metoda využívá aplikace typu server bežícího na centrálním serveru, který naslouchá na daném portu a registruje zařízení, ze kterých byl vyslán požadavek na registraci k serveru.

2.8 Návrh systému

Po konzultaci s vedoucím práce byla navrhnuta základní architektura systému, kterou můžete vidět na obrázku 2.2. Centrem systému je server, který běží na doméně livs.fit.svut.cz. Na tomto serveru je umístěn webový portál, který obsluhuje celý systém. Webový portál obsluhuje systém pomocí skriptů, které jsou zde umístěné a umožňují komunikaci s celým systémem. Pomocí těchto skriptů se provádějí akce, jako například spouštění aplikace na zařízení nebo přidání zařízení do databáze. Centrální MySQL databáze je také umístěna na serveru. V této databázi dochází k výměně informací o zařízeních mezi portálem a systémem. Jsou zde umístěné informace důležité pro nahrávání aplikací do zařízení, popis zařízení a stav zařízení. Z bezpečnostních důvodů je přístup k této databázi možný jen lokálně ze serveru. K databázi tedy nemají přístup samotná zařízení.

K centrálnímu serveru se připojují roboti NAO a ostatní linuxová zařízení. Komunikace mezi těmito zařízeními a serverem probíhá pomocí společné sítě WiFi. Na těchto zařízeních je spuštěn SSH server, ke kterému se připojuje centrální server nyní v roli klienta, a obsluhuje tato zařízení. Zařízení se

zároveň také připojují k samotnému serveru, vždy po spuštění zařízení (v případě robota NAO vždy když dojde ke ztrátě spojení), nebo připojení Arduina. Toto spojení z důvodu bezpečnosti není realizováno přes SSH, ale pouze pomocí TCP. Na centrálním serveru je spuštěn server, který naslouchá příchozím spojením na daném portu a pokud je na zařízení spuštěn klient pro registraci zařízení, zapíše předané informace o zařízení do databáze. Zařízení tedy nemají přístup přes SSH k serveru.

K těmto zařízením se připojují pomocí USB Arduino zařízení. Arduino musí být připojeno k zařízení, které obsahuje software pro jeho obsluhu. Většinou se v praxi bude jednat o Raspberry Pi s připojeným USB rozbočovačem, ale systém umožňuje připojit Arduino k jakémukoliv zařízení i robotu NAO (musel by se však na něj nainstalovat software pro jejich obsluhu). Na zařízení s připojenými Arduiny je klient, který podobně jako klient pro linuxová zařízení, registruje zařízení na serveru. Tento klient je spouštěn pomocí udev.

2.9 Procesy v systému

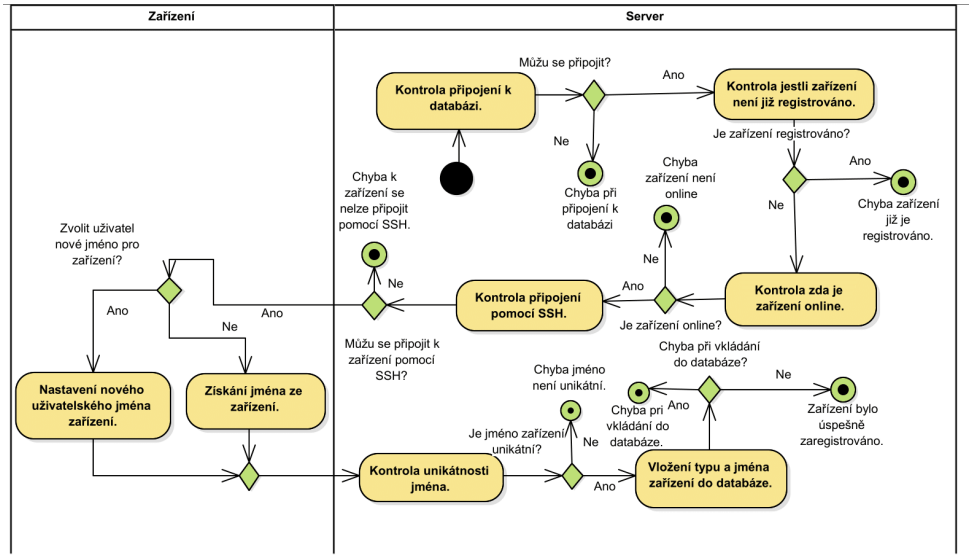
V systému se vyskytuje mnoho procesů, většina z nich však má podobný průběh. Zde uvádím tři procesy které se využívají v systému a na jejichž základě lze přibližně odvodit ostatní procesy v systému.

2.9.1 Vložení zařízení do databáze

Při vkládání do databáze je třeba zajistit přístup zařízení a poslat zařízení veřejný klíč k autentizaci. Tento krok bohužel nemůže být proveden z prostředí webového portálu, protože je třeba zadat heslo k zařízení ručně. Při samotném vkládání je potřeba nejprve zjistit, zda má uživatel přístup do databáze. Následně je provedena kontrola, jestli již v databázi není zařízení zaregistrováno. Poté je je provedena kontrola, zda je zařízení dostupné ze serveru. To se zkouší pouze při vkládání zařízení, neboť nedostupnost odhalí i další krok, zde tedy slouží spíše k diagnostice. Následující krok je již zmíněný test připojení pomocí SSH. Pokud uživatel zadal název, pod kterým chce zařízení registrovat do databáze, pak je tento název nastaven zařízení. Jinak se název získá ze zařízení. Následuje kontrola, jestli je název zařízení unikátní a pokud je, pak se vloží zařízení do databáze. Celý proces je vidět na obrázku 2.3.

2.9.2 Připojení Arduino zařízení k systému

Po připojení Arduina je pomocí udev automaticky spouštěn klient, který registruje zařízení na serveru. Klientu jsou předány informace o zařízení, důležité pro registraci zařízení na serveru a získání jména, pod kterým je uvedeno v adresáři `/dev`. Klient zašle tyto informace serveru, který naslouchá na centrálním serveru. Tento server spustí příslušný skript pro registraci stavu zařízení a ten v databázi vyhledá podle předaných informací jméno zařízení. Jestliže



Obrázek 2.3: Proces registrace zařízení.

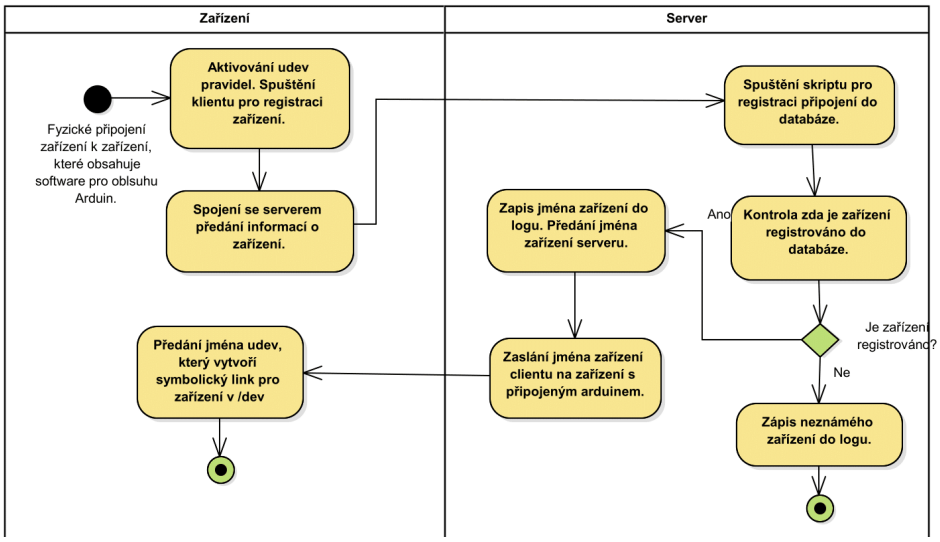
jméno nenalezne, znamená to, že zařízení neprošlo první registrací. Takové zařízení zapíše do logu a proces končí. Pokud jméno nalezne, vrátí ho serveru a ten pošle klientovi. Klient vrátí jméno udev, a ten vytvoří symbolický název zařízení v `/dev`. Celý proces je vidět na obrázku 2.4.

2.9.3 Spuštění programu na zařízení

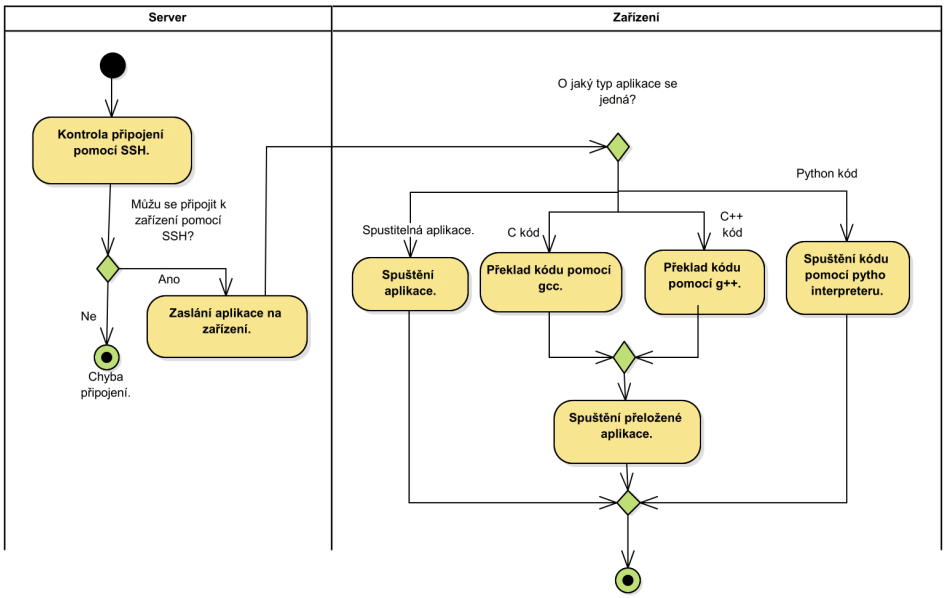
Při spouštění aplikace na zařízení nejprve proběhne kontrola připojení k zařízení pomocí SSH. Následně je na zařízení přenesena aplikace (nebo kód aplikace). Podle typu aplikace je spuštěná požadovaná akce. V případě, že se jedná pouze o spustitelnou aplikaci, pak je spuštěna. Pokud se jedná o kód, pak je přeložen a spuštěn. V případě C kódu překladačem gcc, v případě C++ pomocí g++. Při spouštění python kódu je interpretován python interpretrem. Celý proces je vidět na obrázku 2.5.

2.10 Návrh databáze

Databáze pro systém byla vyvíjena společně s Danem Zatloukalem. Databáze pro systém tvoří jen část celé databáze (vyjmutá část z databáze portálu na obrázku 2.6). Vetší část tvoří databáze samotného portálů. Výhodou společné databáze je rychlý přístup k datům a jednoduché předávání informací mezi systémem a portálem. Na druhou stranu to znamená jisté bezpečností riziko. Databáze obsahuje tyto entity:



Obrázek 2.4: Proces vložení zařízení.



Obrázek 2.5: Proces spuštění aplikace na zařízení.

- **device_model**

  Tato entita reprezentuje model zařízení. Její atributy jsou název modelu zařízení, typ modelu (arduino device / linux device) a popis modelu zařízení, kde mohou být zaznamenány pro uživatele důležité informace o modelu zařízení.

- **production_device_model**

  Tato entita představuje rozšíření modelu zařízení o informace popisující model Arduino zařízení. Zde jsou uloženy informace o zařízení důležité pro nahrávání firmwaru do zařízení. Je zde uložené vendorID, značící identifikátor výrobce zařízení, a productID, které značí identifikátor modelu zařízení. Oba identifikátory mají velikost 16 bitů a jsou zapsány hexadecimálně. MCU označuje typ mikroprocesoru, který obsahuje daný typ zařízení. Programer označuje typ programátoru, který bude použit při nahrávání firmwaru do zařízení pomocí avrdude. Název se řídí dle konvencí avrdude. Baud rate znamená počet změn stavu při přenosu za jednu vteřinu. Při nahrávání určuje rychlost, jakou se komunikuje s programátorem zařízení. Account name značí název účtu, pod kterým jsou zařízení připojena. Používá se v případě, když uživatel neuvede celou adresu zařízení i s uživatelským jménem, ale jen IP adresu.

- **device**

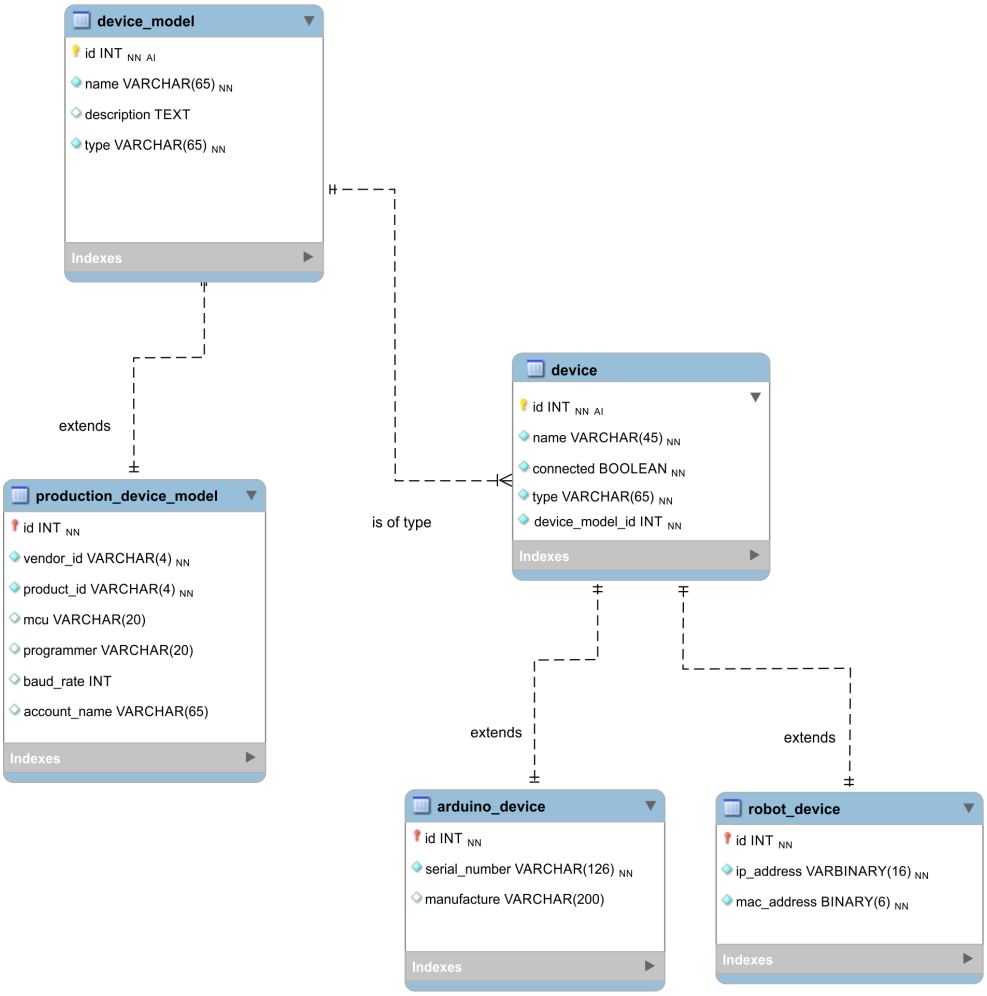
  Tato entita popisuje linuxová a Arduino zařízení. Prvním atributem je název zařízení, který si uživatel zvolil a pod kterým lze nahrávat software do zařízení. Druhým atributem je indikátor stavu zařízení. Jeho hodnota je boolovská, true pro online, false pro offline. Třetím je typ zařízení (arduino device / linux device) a poslední je identifikátor modelu zařízení.

- **arduino_device**

  Tato entita poskytuje informace o Arduino zařízení. Je zde uvedeno sériové číslo zařízení (serial_number), které slouží k identifikaci zařízení a výrobce zařízení (manufacture), výrobce zařízení je uveden zde a ne u production_device_model z důvodu možnosti existence klonu zařízení stejného typu jen od jiného výrobce.

- **robot_device**

  Tato entita popisuje robota (nebo linuxové zařízení). První atributem je IP adresa zařízení ve formátu varbinary, který může obsahovat IPv6 adresu. Druhým atributem je Mac adresa rozhraní, pomocí kterého je zařízení připojené k internetu, v hexadecimálním zápisu.



Obrázek 2.6: Databáze systému.

Kapitola 3

Realizace

Pro spojení webového portálu se systémem se používají bash skripty, které jsou hlavním cílem realizační části práce. Pro zajištění infrastruktury systému se používá dvojice serveru a klientu, která se stará o registraci připojených zařízení do systému. Systém obsahuje několik verzí klientu specifických pro dané zařízení. Klient i server jsou napsány v jazyce python. Dále jsou také vytvořena pravidla udev pro připojení Arduin. A poslední částí je realizace instalačních balíčků pro určité typy zařízení. Popis použití přepínačů je uveden v uživatelské příručce, případně lze vyvolat nápovědu pro skripty použitím přepínače -h.

3.1 Společné mechanismy skriptů

V sytému se vyskytuje několik mechanismů, které jsou společné více skriptům, a proto bych rád popsal tyto mechanismy odděleně a následně se již jen odkazoval na tento popis.

3.1.1 Přihlášení do MySQL databáze

Tento příkaz 1 slouží k přihlášení do databáze pomocí mysql klienta. Prvními třemi proměnnými jsou přepínače pro přihlášení do MySQL databáze. Tyto proměnné obsahují zároveň přepínač i hodnotu a to z toho důvodu, aby nedocházelo ke složitému větvení programu. Pokud jsou uvedeny obě varianty přihlášení, klient se pokusí přihlásit oběma variantami. Proměnná LOGIN_PATH obsahuje přepínač `--login-path=` a název připojení login path. Proměnná PASSWORD obsahuje přepínač `-p` a heslo k databázi. Proměnná USER_NAME obsahuje přepínač `-u` a uživatelské jméno k databázi. Proměnná HOST obsahuje přepínač `--host=` a IP adresu databáze. Většinou je její hodnota local_host. Přepínač `-D` slouží ke specifikování jména databáze, ke které se má klient připojit. Přepínač `-s` spustí „tichý chod“ příkazu, který zajistí, aby se do výstupu nevypisovaly názvy a nevykreslovaly obrysy tabulek. Po-

sledním přepínačem je `-e`. Ten zajistí vykonání SQL příkazu, který následuje za ním. Tento příkaz je uveden v uvozovkách. Zde je uveden ilustrační příkaz, který vypíše počet Arduino zařízení s daným sériovým číslem.

```
mysql ${LOGIN_PATH} ${PASSWORD} ${USER_NAME} ${HOST}
↪  -D${NAME_OF_DATABASE} -se  "SELECT COUNT(*) FROM
↪  arduino_device WHERE serial_number = '${SERAILID}'"
```

Výpis kódu 1: Spuštení MySQL příkazu klientem mysql

3.1.2 Přípojení k zařízení pomocí SSH

Tento příkaz 2 slouží k přípojení se k SSH serveru, který je spuštěn na zařízení. Prvním přepínačem je `-i`, který slouží k určení umístnění soukromého klíče. Umístnění SSH klíče je uložené v proměnné SSH_PRIVATE_KEY. Tento klíč je použit pro autentizaci k zařízení. Dalším přepínačem je `-o`, který slouží k modifikaci nastavení ssh klienta. Zde je nastaven časový limit pro navázání spojení se zařízením a také je zde zapnutý „BatchMode“, který zajistí, aby v případě, že selže autentizace pomocí klíčů, nedošlo k výzvě zadání uživatelského hesla pomocí uživatele. Dále následuje uživatelské jméno uživatele, na jehož účtu se provede vykonání příkazu, a IP adresa zařízení. Poté následuje v uvozovkách příkaz k vykonání.

```
ssh -i ${SSH_PRIVATE_KEY} -o BatchMode=yes -o
↪  ConnectTimeout="${SSH_TIMEOUT}"
↪  "${ROBOT_USERNAME}"@"${DEVICE_IP}"  "echo connect"
```

Výpis kódu 2: Spuštení příkazu na vzdáleném serveru klientem ssh

3.1.3 Přenos souborů pomocí rsync

Pro přenášení souborů na zařízení je využit program rsync v tomto nastavení 3 . Ve skriptech, které slouží výhradně pro přenos souborů, může být použito modifikované nastavení. První přepínač `-e` slouží k modifikaci ssh, který rsync využívá. Zde se nastavuje umístnění soukromého klíče. Pomocí přepínače `--timeout` se nastaví časový limit pro připojení k zařízení. Archivní mód se zapíná přepínačem `-a`. Archivní mód při kopírování použije rekurzi, zachovává oprávnění a symbolické linky. Při použití přepínače `-z` se přenášená data komprimují. A přepínač `-P` zobrazí postup přenášení souborů a zároveň zachová částečně přenesené soubory. Dále následuje soubor pro přenos a cíl, kam se má soubor přenést. Cíl se skládá z uživatelského jména odděleného od IP adresy zavináčem a cílovou složkou na zařízení oddělenou dvojtečkou.

```
rsync -e "ssh -i ${SSH_PRIVATE_KEY}"
↪  --timeout="${RSYNC_TIMEOUT}" -azP "${APP_FILE}"
↪  "${ROBOT_USERNAME}"@"${DEVICE_IP}":
↪  "${DIRECTORY_FOR_PROGRAMS}/"
```

Výpis kódu 3: Přenos souborů pomocí rsync v nastavění pro bežný přenos

3.2 Registrace zařízení do systému

Pro možnost nahrávání aplikací do zařízení podle jména zařízení nebo zobrazení stavu zařízení je nutné zařízení registrovat do databáze. Základní obsluha zařízení je možná i bez registrace. Při registraci zařízení je automaticky vytvořen typ zařízení (pokud již neexistuje), k němu je vytvořen jeho popis a následně je zařazeno do databáze samotné zařízení.

3.2.1 Zaslání veřejného SSH klíče zařízení

Před přidání linuxového zařízení do databáze je nutné zaslat veřejný klíč zařízení. K tomu slouží skript first_login.sh, který pošle zadaný veřejný klíč zařízení. K tomu se využívá příkaz `ssh-copy-id`. Tento příkaz zašle zadaný veřejný klíč ssh serveru (tedy zařízení) a uloží ho ve souboru authorized_keys. Ten je umístěn v adresáři s lokálním nastavením uživatele (výchozí umístnění je `~/.ssh`) a obsahuje typ klíče, klíč samotný a název uživatelského účtu, ze kterého byl klíč odeslán. Po spuštění skriptu je nutné zadat heslo k zařízení a následně dojde k uložení otisku zařízení do souboru known_hosts umístěného v lokální konfigurační složce serveru.

Tento skript bohužel nejde spustit z webového portálu, neboť vyžaduje zadání hesla uživatelem. V testovací verzi systému bylo možné zaslat klíč pomocí příkazu sshpass, který předal heslo příkazu `ssh-copy-id`. Bohužel tento příkaz není kompatibilní se systémem FreeBSD, který je nainstalován na serveru. Další možností, jak předat heslo by bylo použití příkazu expect, který slouží k automatickému vyplňování žádostí o vstup od programů. Tento program také není plně podporován na serveru.

3.2.2 Registrace linuxového zařízení

Pro registraci linuxového zařízení slouží skript insert_robot.sh. Podrobně je možné si proces registrace zařízení prohlédnout na obrázku 2.3. Skript zkontroluje přístup do databáze, pomocí příkazu ping zjistí, zda je zařízení online, zkontroluje jestli zařízení není registrováno a také přístup k zařízené pomocí SSH. Poté přečte jméno a typ zařízení (pokud je nastavena volba změny jména, tak jej nastaví) a zkontroluje unikátnost jména. Dále přečte MAC adresu zařízení ze souboru `/sys/class/net/*interface_name*/address`, ve výchozím nastavení z rozhraní wlan0. Bohužel nelze triviálně získat z IP

adresy zařízení MAC adresu zařízení. Vzhledem k tomu, že se zařízení nemusí nacházet ve stejné síti jako je server, nelze využít ARP dotaz. Další možností jak spárovat IP adresu s MAC adresou, je výpis příkazu ifconfig. Ten je však platformě závislý a navíc některé distribuce jej nepodporují (např. v debianu byl nahrazen příkazem ip). Nyní má skript veškeré důležité informace a může vložit zařízení do databáze. Pro vložení do databáze se používají funkce MySQL. IP adresa zařízení je do databáze ukládána pomocí mysql funkce `INET6_ATON('${DEVICE_IP}')` tato funkce převede IP adresu (string) do binární podoby. Funkce převádí jak IPv4, tak i IPv6 adresy. Pro převod zpět se používá funkce `INET6_NTOA(ip_address)` Pro převod MAC adresy do hexadecimální podoby stačí napsat před proměnou x viz `x'${ROBOT_MAC_ADDRESS}'`, zpět pomocí funkce `HEX(mac_address)` Prvně je do databáze vložen typ zařízení (entita device_model), následně popis typu zařízení (entita production_device_model), zařízení (entita device) a popis zařízení(robot_device). SQL příkazy obsahují kontrolu proti vkládání duplicit. Při tomto procesu by tedy neměla nastat chyba, a pokud nastane, znamená to chybu v datech nebo systému.

Možností, která se nedostala do finální implementace z důvodu bezpečnosti, bylo přiřazení jména zařízení a IP adresy do `/etc/hosts`. Záznamy v tomto souboru se chovají podobně jako záznamy v DNS (jedná se však o službu přepínání jmen NSS). Pomocí nich by pak bylo možné přistupovat ze serveru k zařízení pomocí jména zařízení. Řešením je vytvoření vlastního DNS serveru pro překládaní jmen zařízení. Další možností, která se nedostala do finální implementace, je možnost vytvoření autentizace zařízení pro připojení přes SSH k centrálnímu serveru. Zde by hrozilo odcizení soukromého klíče ze zařízení a následné přihlášení k serveru útočníkem. Tomu by šlo zabránit vytvořením heslové fráze zařízení, kterou je nutné zadat před použitím soukromého klíče. Heslovou frázi lze uložit do paměti, aby ji nebylo nutné zadávat pomocí ssh agentu, ale po restartu je nutné zadat heslovou frázi znovu a vzhledem k tomu, že zařízení se restartují často, bylo toto řešení shledáno nedodatečné.

3.2.3 Registrace Arduina

Pro registraci Arduino zařízení do databáze slouží skript insert_arduino.sh. Tento skript podle jména zařízení, které bylo přidělené zařízení od systému se nachází v `/dev` (typicky ttyACM* nebo ttyUSB*) a vyhledá informace o zařízení a uloží je do databáze spolu s přezdívkou zařízení. Tento skript se připojuje k zařízení s připojeným Arduinem a je třeba specifikovat IP adresu zařízení. Skript nejprve zkontroluje připojení k databázi, unikátnost přezdívky zařízení a připojení k zařazení. Poté spustí na zařízení nástroj udevadm v této podobě 4, který přečte VENDORID, PRODUCTID, SERAILID, MANUFACTURE a PRODUCTNAME zařízení. Následně zkontroluje podle unikátnosti sériového čísla, zda již nebylo zařízení registrováno. Poté dojde k vložení informací o zařízení do databáze. Nejprve se vloží typ zařízení (entita

```
udevadm    info -a -n "${DEVICEPATH}" info udev
```

Výpis kódu 4: Zobrazení atributů zařízení pomocí udevadm

device_model), následně popis typu zařízení (entita production_device_model), zařízení (entita device) a popis zařízení (arduino_device). SQL příkazy obsahují kontrolu proti vkládání duplicit. Při tomto procesu by tedy neměla nastat chyba, a pokud nastane, značí to chybu v datech nebo systému.

Před samotným přidáním Arduino zařízení do databáze, zvláště pak jedná-li se o typ zařízení, který ještě nebyl v zařízení registrován, je vhodné zkontrolovat, zda jeho USB popisovače obsahují všechny důležité informace pro registraci, tedy zejména sériové číslo, identifikátor výrobce a produktu. V případě registrace zařízení, které tyto identifikátory neobsahuje, může dojít k převzetí informací o zařízení z rodičovského zařízení (tedy např. usb rozbočovače), dojde tak ke špatné identifikaci zařízení.

3.3 Spuštění aplikace na zařízení

Aplikace na zařízení se spouští pomocí příkazu run_on_robot.sh. Průběh procesu spouštění aplikace na zařízení je vidět na obrázku 2.5. Tento skript podporuje spouštění C a C++ kódů, python skriptů a spustitelných aplikací (bash skript, přeložená binární aplikace, ...). Skript po spouštění ověří připojení k zařízení pomocí SSH a pošle zařízení aplikaci do složky s aplikacemi (ve výchozím nastavení `~/livs/DIRECTORY\_FOR\_PROGRAMS/`). Následně dojde v případě spustitelné aplikace k přímenému spuštění, v případě python skriptu k interpretaci pomocí python interpretru a v případě C++ kódu je kód přeložen pomocí g++ s přepínačem `-std=c++11` do souboru název_kódu.out a následně spuštěn. Stejně se postupuje s C kódem, jen je použitý překladač gcc. Po přesunutí souboru na stranu zařízení je důležité nezapomenout odstranit název cesty z názvu aplikace.

V současné době robot NAO nepodporuje překlad C a C++ kódu.

3.4 Nahrání firmwaru do Arduino zařízení

Pro nahrávání hex souborů do Aruino zařízení slouží skript load_to_arduino_from_server.sh. Tento skript spustí program avrdude na zařízení, ke kterému jsou Arduina připojena a pomocí něj do nich nahraje požadovaný firmware.

Skript nejprve zkontroluje SSH připojení k zařízení s připojenými Arduiny, následně zkusí vyhledat Arduino zařízení pomocí jeho názvu, zda je registrované. Pokud je zařízení registrované, vyhledá o něm informace v databázi a následně nastaví parametry pro avrdude. V případě, že zařízení není registrované v databázi, pokusí se skript vyhledat, zda není registrován alespoň typ

zařízení a vyhledá podle VendorID a ProductID typ zařízení. Pokud není nalezen typ zařízení, použije se nastavení dané přepínači. Pokud toto nastavení není specifikováno pomocí přepínačů, skript skončí chybou. Skript používá avrdude s temito parametry 5.

```
avrdude  -q -p  ${MCU} -c ${PROGRAMER}  -P /dev/${DEVICE} -b
↪  ${BAUD_RATE} -D -U flash:w:${FILE_PATH}:i
```

Výpis kódu 5: Nahrání hex kódu do Arduino zařízení

- `-q` se zapíná tichý mód.
- `-p` slouží k výběru typu mikroprocesoru, název se řídí specifikací avrdude.
- `-c` slouží k výběru typu programátoru zařízení, název se řídí specifikací avrdude.
- `-P` specifikuje cestu k zařízení.
- `-b` určuje baud rate pro komunikaci s deskou.
- `-D` zabrání automatickému smazaní flash paměti před nahráním.
- `-U` slouží k operacím s pamětí zařízení. Za ním následuje nastavení prováděné operace, kde jsou jednotlivé parametry oddělené dvojtečkou. První parametr značí typ paměti pro nahrávání, v tomto případě flash, dále se specifikuje typ paměťové operace, tedy w pro zápis. Dalším parametrem je název souboru k zápisu a posledním je typ souboru. Zde je typem souboru intel Hex, tedy parametr i.

3.5 Instalace aplikace na zařízení

Instalace aplikace na zařízení se provádí pomocí skriptu install_app.sh. Tento skript slouží k instalaci pouze na roboty NAO nebo zařízení s linuxovou distribucí Gentoo. Skript prvně zkontroluje připojení k zařízení pomocí SSH a následně pošle zařízení instalační soubor. Instalační soubor je rozbalen v jeho kořenovém adresáři pomocí příkazu 6. Tento příkaz používá přepínač -x pro rozbalení souboru, -f pro právi se soubory a -j pro typ archivu bzip2. Cílem pro rozbalení je složka, použije se tedy přepínač -C. a tím dojde k instalaci

```
sudo tar xjf ${DIRECTORY_FOR_INSTALL_FILE}/${INSTALL_FILE} -C /
```

Výpis kódu 6: Rozbalení instalační aplikace na zařízení

aplikace. Instalační soubor musí mít tar.gz2 formát. Důležité je, aby uživatel měl potřebná oprávnění k zápisu do kořenového adresáře.

3.6 Restart zařízení

Restart zařízení se provádí pomocí příkazu restart_device.sh. Tento příkaz zkontroluje připojení k zařízení pomocí SSH a následně spustí na zařízení příkaz, reboot pokud uživatel zvolil volbu pro restart zařízení, nebo halt, pokud uživatel chce zařízení vypnout. Důležité je, aby uživatel měl oprávnění spouštět tyto příkazy.

3.7 Záloha zařízení

Zálohu zařízení a přenos souborů ze zařízení provedeme pomocí skriptu backup_robot.sh. Tento skript prvně zkontroluje SSH připojení k zařízení a následně přenese pomocí rsync požadované data. V tomto skriptu je použití rsync modifikováno. Příkaz rsync je použit v následující podobě 7. Je zde přidán přepínač `-t` pro zachování času modifikace přenášených souboru. Toto nastavení může ušetřit objem přenášených dat v případě, že není použité porovnávání souborů pomocí kontrolního součtu, ale pouze podle času modifikace. Dále jsou zde použité přepínače `-h, -v` a `--stats` pro popis akcí prováděných pomocí rsync a zobrazení statistiky přenosu. Tento výstup může být použit na webovém portálu pro zobrazení stavu přenosu. Při vytváření zálohy je také důležité vynechat některé umístnění. K tomu slouží přepínač `--exclude=`, který zařídí, aby byly vynechány soubory jejichž kopírování nedává smysl, nebo by se zacyklilo. Pro přenos je možné přepínačem `-c` zapnout možnost porovnávání souborů na základě kontrolních součtů (více o této metodě).

```
rsync -e "ssh -i ${SSH_PRIVATE_KEY}"
↪  --timeout="${RSYNC_TIMEOUT}" -"${CHECKSUM}"havztP
↪  --exclude={"/dev/*", "/proc/*", "/sys/*", "/tmp/*",
↪  "/run/*", "/mnt/*", "/media/*", "/lost+found"}
↪  "${ROBOT_USERNAME}"@"${DEVICE_IP}":"${BACKUP_FOLDER}"
↪  "${OUTPUT_FOLDER}"
```

Výpis kódu 7: Záloha zařízení pomocí rsync

Možností, který tento skript nabízí je vytváření záplat pro zařízení a tedy snadnou distribuci updatu. Pro vytvoření je nutné prvně vytvořit zálohu zařízení před aplikováním změn. Poté jsou provedeny změny na zařízení a následně je zařízení znovu zazálohováno do jiné složky, akorát je při zálohování

uvedena složka pro srovnání změn (složka s první zálohou zařízení). Tato volba se u rsync zapíná pomocí `--compare-dest=`.

3.8 Poslání souboru do zařízení

Zaslání souboru do zařízení nebo obnovu zařízení provedeme pomocí skriptu send_to_robot.sh. Tento skript funguje podobně jako skript pro zálohu zařízení, jen je zde prohozen cíl a zdroj dat. Tento skript přináší možnost obnovy zařízení z bodu obnovy pomocí přepínače `--delete` pro rsync. Tento přepínač smaže veškeré nadbytečné soubory v cílové složce, které nejsou umístěné ve zdrojové složce. Tento způsob je nutné provádět s obezřetností, neboť hrozí ztráta dat.

3.9 Zjištění online zařízení

Pro webový portál je důležité, aby systém měl přehled, které ze zaregistrovaných zařízení je online. Podle toho pak může nabídnout uživateli přehled o zařízeních, která jsou připravená pro uživatelovu akci. V systému existují dva způsoby pro zjištění, jestli je linuxové zařízení online. První je pomocí jednoduchého IP adres skenu. Druhý způsob je zapsání zařízení do databáze při spuštění nebo po výpadku sítě. Tento způsob nezjistí, zda je zařízení offline.

Pro Arduino zařízení existuje jen jeden způsob, a to je registrace zařízení na serveru při připojení nebo odpojení zařízení.

3.9.1 Online sken

Online scan se spouští pomocí skriptu check_online_robots.sh. Tento script je spouštěn pomocí cron v nastaveném časovém intervalu závislém na počtu zařízení. Skript může být také spuštěn manuálně z webového portálu. Tento skript z důvodu, že je spouštěn automaticky, získává přihlašovací údaje k databázi z konfiguračního souboru webového portálu nebo případně (pokud je definováno) z login_path. Prvně zkontroluje přístup k databázi a následně vybere IP adresy všech zařízení. Následně se pokusí spojit z každým zařízením pomocí příkazu ping 8. Tento příkaz používá přepínač -c pro nastavení počtu pokusů o spojení se zařízením.

```
ping -c "${NUMBER_OF_ATTEMPTS}"  "${line}"
```

Výpis kódu 8: Test připojení zařízení pomcí příkazu ping

Pokud se spojení s zařízením povedlo, zapíše do databáze, že zařízení je online, v opačném případě že je offline.

3.9.2 Registrace zařízení na serveru

Na centrálním serveru je spuštěn python program server.py. Tento program registruje příchozí spojení na zadaný port. Program využívá pakety typu TCP. Program nejprve přečte konfigurační soubor a podle něj otevře port na serveru (ve výchozím nastavení port 2500). Na tomto portu naslouchá příchozí spojení od jednotlivých klientů zařízení. V případě příchozího spojení vytvoří nový proces. Tento proces nejprve přečte zprávu a na základě prefixu zprávy spustí příslušný skript pro zapsání zařízení do databáze. Prefixy zprávy závisí na konfiguraci, ale ve výchozím nastavení jsou -ri pro připojení linuxového zařízení, -ai pro připojení Arduino zařízení a -ao pro odpojení Arduino zařízení. Při předávání zprávy je zpráva zbavena prefixu. Spuštěné skripty obsahují automatické přihlášení do databáze.

Pro připojení linuxového zařízení je spuštěn skript robot_login_sf.sh s dvěma argumenty. Prvním argumentem je jméno zařízení a druhým je IP adresa zařízení. Skript se pokusí podle jména zařízení v databázi vyhledat zda je zařízení registrováno. Pokud ano, zapíše do databáze, že zařízené je online. Následně zařízení zapíše informaci o připojení zařízení do logu. Výchozí název souboru pro log je plug.log. Formát log záznamu je „NAME:*název_zařízení* IP_ADDRESS:*IP_adresa_zařízení *čas_ připojení* was plug in“. Pokud není zařízení registrováno v databázi, je ještě v logu připojen dovětek „unknown device“.

Pro připojení Arduino zařízení je spuštěn skript plug_in_script.sh s jedním argumentem. Ten představuje zpráva, kterou obdržel server. Tato zpráva má předem definovaný formát. Obsahuje čtyři hodnoty oddělené pomocí znaku „^“. Prvním hodnotou je sériové číslo, následuje identifikátor výrobce VendorID, identifikátor modelu zařízení ProductID a název výrobce. Následně je podle sériového čísla zjištěno, jestli je již zařízení registrováno do databáze. Pokud je registrováno, vyhledá se jeho přezdívka. Ta je následně předána zpátky serveru přes standardní výstup. Také se zapíše, že zařízení je online, a zapíše se do logu. Formát pro log je „Serial number:*sériové_číslo* VendorID: *identifikátor_várobce* ProductID: *identifikátor modelu zařízení* Manufacture: *název výrobce zařízení* *čas_připojení* was plug in“. Stejně jako v případě linuxového zařízení je v případě, že zařízení není registrováno vložen dovětek „unknown device“.

Pro odpojení Arduino zařízení je spuštěn skript plug_out_script.sh s jedním argumentem. Tento skript poskytuje podobnou funkcionalitu jako skript pro připojení. Jen nevrací jméno zařízení, ale do databáze zapisuje, že zařízení je offline a v logu je napsáno za časem připojení was plug out.

Server spouští skripty pomocí dvou funkcí, v závislosti, zda potřebuje návratovou hodnotu (je třeba při volání skriptu plug_in_script, který vrací přezdívku zařízení).

Po zpracování zprávy server odešle potvrzující zprávu „OK“. Pokud se jedná o odpověď na připojení Arduina, tak místo potvrzující zprávy server

```
p = Popen(['./plug_in_script.sh',
↪  str(data,"utf-8")[len(ARDUINO_IN): -len (END_OF_MESSAGE)]],
↪  stdin=PIPE, stdout=PIPE, stderr=PIPE)

subprocess.check_call(["./robot_login_sf.sh", str(data,"utf-8")
↪  [len(ROBOT_IN):-len(END_OF_MESSAGE)],str(addr).split("'",
↪  1)[1].split("'", 1)[0]])
```

Výpis kódu 9: Fuknce pro spouštění bash skriptů z python programu

zasílá přezdívku zařízení (tato přezdívka slouží k pojmenování zařízení více). Následně proces končí a čeká se na další spojení.

3.9.2.1 Připojení Arduino zařízení

Pro registraci připojení Arduino na server se používá python program client_arduino.py. Tento program je umístěný na zařízení s připojenými Arduiny a je spouštěn pomocí udev. Ke spuštění dochází při připojení nebo odpojení Arduino zařízení. Tento klient je spouštěn s jedním argumentem, který obsahuje čtyři hodnoty oddělené pomocí znaku stříška – „^“, a prefixem značící operaci se zařízením, která byla provedena. První hodnotou je sériové číslo, následuje identifikátor výrobce VendorID, identifikátor modelu zařízení ProductID a název výrobce. Prefix obsahuje -ai pro připojení Arduina a -ao pro odpojení Aruina. Program nejprve načte konfigurační soubory a podle nich se připojí k serveru. K přípojení potřebuje především IP adresu serveru a číslo portu. Po přípojení k serveru je zaslána zpráva, její obsah je první argument. Poté se počká na přijetí zprávy od serveru s přezdívkou zařízení, které se předá přes standardní vstup udev.

3.9.2.2 Připojení robota NAO

Pro robota NAO byl napsán speciální python klient client_state.py pro spojení se serverem. Tento klient využívá služeb frameworku NAOqi. Tento klient umožňuje navíc od ostatních klientů znovu registrování robota na serveru po výpadku spojení se serverem. Spouštění tohoto klientu je nastaveno pomocí zápisu do souboru autoload.ini, který se nachází v adresáři `~/naoqi/preferences/`. Soubor umožňuje načtení knihoven (.so), spuštění python skriptu a nebo spuštění programu. Tento soubor je načten při spuštění NAOqi. Po spuštění klientu by měly být předány argumenty k připojení ke správnému brokeru. Těmito argumenty jsou IP adresa a port brokeru. Bohužel při automatickém spouštění programu dojde z neznámého důvodu k pádu programu při volání metody `parser.parse_args()`. Proto je nastavena IP adresa a port brokeru napevno a hodnoty lokálního brokeru 127.0.0.1 a port 9559.

Broker slouží k registraci knihoven a modulů a jsou přes něj tyto moduly dostupné.

Klient využívá pro svojí funkci události. Událost vyžaduje vytvoření vlastního modulu, který je zaregistrován v Brokeru, a poté, co událost nastane, je zavolána metoda modulu, která byla přiřazena dané události. Klient vytváří modul ReactToNetworkState s metodou onChanged, která je volána jako reakce na událost. Událost je typu NetworkStateChanged. Tato událost nastává, pokud dejde ke změně v připojení k síti. Stav připojení k síti má tři stavy online, ready a offline. Po každé změně dojde k zavolání události. Událost se registruje u brokeru pomocí metody `subscribeToEvent()`, z modulu memory. Tato metoda vyžaduje tři parametry. Prvním je typ události, druhým je je vytvořený modul a třetí metoda, která se volá poté, co nastane událost. Této metodě jsou předány hodnoty události. U tohoto typu události je předáno jméno události, stav připojení a identifikátor registrace události. Pro odregistrování reakce na událost slouží metoda `unsubscribeToEvent()`, která vyžaduje dva parametry. První je název události a druhý je název modulu.

Po spuštění klienta dojde nejprve k připojení brokeru. Následně je inicializovaný modul ReactToNetworkStatem, vytvořením instance stejnojmenné třídy. Tato instance musí být přístupná globálně, proto je použito klíčové slovo global. Při inicializaci třídy dojde k načtení modulu ALTextToSpeech, který umožní, aby NAO přečetl požadovaný text. A také dojde k zaregistrování události. Následně klient čeká na to, až nastane událost v nekonečné smyčce. V této smyčce je umístěna metoda sleep s parametrem 1, aby nedoházelo ke zbytečnému vytěžování procesoru, a také je zde umístěno zachytávání přerušení od klávesnice, které ukončí klienta, pokud by byl klient spouštěn ručně. Pokud je vytvořena událost, dojde k zavolání metody `onChanged()`. Ta nejprve odregistruje událost, aby nedošlo znovu k zavolání události během zpracování této události. Poté v případě, že stav události je online, zašle zprávu se jménem zařízení a prefixem -ri k registraci serveru. Jméno zařízení získá pomocí funkce `robot_name()` z balíčku get_robot_name. Poté čeká na odpověď od serveru. Pokud je odpověď „OK“, NAO řekne: „Pripojen k serveru *IP adresa serveru*“. Na konci metody musí být znovu zaregistrována událost, stejně tak, jako když dojde k chybě spojení a metoda je ukončena předčasně.

Tento způsob předpokládá, že robot pro připojení k internetu a k serveru využívá stejnou síť. V případě, že by tomu tak nebylo (byl by připojen k serveru např. Ethernet kabelem), poté by bylo vhodnější využívat událost typu NetworkServiceStateChanged, která umožňuje sledovat stav dané služby připojení. Současná metoda má ale výhodu, že se nemusí specifikovat síť, ke které je robot připojen.

3.9.2.3 Připojení ostatních linuxových zařízení

Pro registraci připojení linuxových zařízení se používá program client.py, který je spouštěn pomocí cron. Pro registraci spuštěním aplikace pomocí cron je

třeba zapsat zařízení do cron tabulky. Pro spuštění programu po spuštění systému slouží znak `@reboot`. Dále se uvede spouštěný program. V tomto případě je před spuštění programu vloženo volání příkazu sleep (s parametrem 10 vteřin), aby byl program spuštěn až po spuštění veškerých služeb, které využívá. Program nejprve načte konfigurační soubory a podle nich se připojí k serveru. Poté pomocí funkce `robot_name()` z balíčku get_robot_name získá název zařízení a pošle zprávu s prefixem -ri a názvem zařízení serveru. Poté počká na přijetí zprávy od serveru. Pokud zpráva obsahuje „OK“, pak pak bylo zařízení úspěšně zaregistrováno. Jinak skončí program chybou.

3.10 Udev pravidla

Pravidla slouží k detekci připojení Arduino zařízení a předělování jmen zařízením. Pravidla jsou umístěná v `/etc/udev/rules.d` na zařízení s připojenými Arduiny. Systém obsahuje dvě pravidla, jedno je 98-device.rules pro připojení zařízení do systému a druhé je 97-device_remove.rules, které se spouští po odpojení zařízení ze systému. Vysoké priority (97,98) s kombinací s tím že pravidla zachycují většinu připojených USB, jsou dány specifickým nasazením na zařízení, které slouží k připojování Arduin. V jiné aplikaci by byly nevhodné. První pravidlo:

```
SUBSYSTEMS=="usb", ACTION=="add", SUBSYSTEM=="tty" ,
↪  ATTRS{idVendor}=="*", PROGRAM+="client_arduino.py '-ai
↪  $attr{serial} ^ $attr{idVendor} ^ $attr{idProduct} ^
↪  $attr{manufacturer}   " ,  MODE+="0666", GROUP+="plugdev",
↪  SYMLINK+="%c"
```

Výpis kódu 10: Pravidlo pro identifikaci zařízení při vložení zařízení

První pravidlo určuje, že subsystém zařízení má být USB a tty. Dále je uvedena akce, na kterou se má pravidlo provést, tedy po připojení zařízení. Možná nadbytečně vypadá požadavek, aby byla zahrnuta všechna zařízení, která obsahují jakékoliv Vendor Id. Pokud by však nebyl tento požadavek zahrnut, mohl by se odkaz, který vytváříme, vytbořit na jinou vrstvu v hierarchii zařízení (pozorováno na Raspberry Pi) a na kterou následně nemáme právo zápisu. Parametr program spustí námi definovaný skript, kterému předáme následující parametry zařízení, a uloží výstup programu (vše co program vypíše do standartního výstupu) do proměnné `%c`, oddělené oddělovači pro pozdější zpracování. Parametr MODE nastaví práva k zařízení, aby bylo možné do zařízení nahrávat soubory a ze zařízení číst (např. pomocí avrdude). Pomocí GROUP je zařízení přiřazeno do správné skupiny také proto, aby k němu byla správná přístupová práva. Poslední parametr je SYMLINK, který získává údaje z proměnné `%c`, ve které je uložen výstup z programu. Tento parametr

nastaví symbolický link na zařízení. Jméno symbolického linku je získáváno pomocí skriptu client_arduino.py z serveru.

```
SUBSYSTEMS=="usb",ENV{DEVTYPE}=="usb_device",
↪  RUN+="client_arduino.py '-ao $env{ID_SERIAL_SHORT} ^
↪  $env{ID_VENDOR_ID} ^ $env{ID_MODEL_ID} ^ $env{ID_VENDOR}'",
↪  ACTION=="remove"
```

Výpis kódu 11: Pravidlo pro identifikaci zařézení při odpojení

Druhé pravidlo obsahuje podobné parametry jako první, jen je zde na místo atributů zařízení ATTRS využito systémových proměnných ENV. A to z toho důvodu, že po odpojení zařízení již nejsou atributy zařízení dostupné a jedinou možností, jak se dostat k informacím o zařízení představují systémové proměnné. Dále místo volání PROGRAM využívá toto pravidlo volání RUN, které je jednodušší a nevrací výstup do proměnné. Run spouští skript, který upozorní server, že zařízení bylo odebráno. Posledním parametrem je ACTION remove, tedy že pravidlo je aktivováno na odpojení zařízení.

3.11 Konfigurační soubory systému

Systém obsahuje dva typy konfiguračních souborů. První je standardní bash konfigurační soubor a druhý je konfigurační soubor pro python, který má vlastní strukturu. Tyto konfigurační soubory umožňují snadnou změnu parametrů systému, jeho přenositelnost do jiné sítě nebo úpravu parametrů zařízení. Jeho struktura je „klíčové_slovo: hodnota“. Řádek, který nezačíná klíčovým slovem, je považován za komentář. Konfigurační soubor pro nastavení skriptů se jmenuje rms_server.conf. Konfigurační soubory pro nastavení python programů se nazývají device.conf a server.conf. Tyto soubory také slouží k nastavení informací o zařízení, proto je zde uvedena jejich struktura. Soubor device.conf obsahuje:

- server_ip: – IP adresa centrálního serveru, ke kterému se připojují zařízení
- server_port: – port, na kterém poslouchá server pro registraci spojení na centrálním serveru
- name: – název zařízení(není u robota NAO)
- buffer_size: – velikost bufferu pro přenos dat TCP pakety
- connection_time_out: – časový limit vytvoření spojení
- type: – typ zařízení Soubor server.conf má stejný obsah jen neobsahuje server_ip.

3.12 Instalační balíčky systému

Systém obsahuje instalační balíčky pro jednotlivá zařízení, které umožňují snadnou instalaci systému na mnoho zařízení daného typu. Byly vytvořeny dva instalační balíčky. Jeden pro robota NAO a druhý pro Raspberry Pi s možností připojení Arduin. První instalační balíček obsahuje instalační soubor install_rms_on_nao.sh, konfigurační soubor device.conf, klienta pro připojení k serveru klient_state.py a python programy pro správu jména zařízení get_robot_name.py a set_robot_name.py. Instalační soubor vytvoří složku pro systém a nastaví automatické spouštění klienta.

Instalační balíček pro Raspberry Pi navíc obsahuje skripty pro obsluhu Arduin a klienta pro komunikaci Arduin se serverem client_arduino.py. Dále jsou zde udev pravidla pro detekci zařízení. Python programy pro změnu a získání jména zařízení jsou jiné než pro robota NAO, ale mají stejný název.

3.13 Souborová struktura systému na zařízeních

Na zařízeních je po instalaci balíčku vytvořena souborová struktura pro systém. Tato struktura umožňuje přehledné umístnění souborů pro systém a ulehčuje přístup ke konfiguračním souborů. Má následující strukturu:

```
livs
├── USER_INSTALLATIONS
└── USER_PROGRAMS
```

Ve složce livs jsou umístněny konfigurační soubory zařízení. Dále skripty pro obsluhu zařízení a také klient pro připojení k serveru. Do složky USER_INSTALLATIONS se ukládají instalační soubory, které byly nainstalovány na zařízení. Do složky USER_PROGRAMS se přenášejí soubory, které se spouští nebo překládají na zařízení a v případě zařízení, ke kterému jsou připojena arduina, jsou zde ukládány také aplikace pro Arduino.

3.14 Budoucí rozvoj systému

Systém umožňuje snadné rozšíření o další funkce na základě již naspaných skriptů a vytvořených procesů. Případně pokud by se ukázalo, že uživatelé potřebují spouštět na robotech velké množství různých jednoduchých příkazů, byla by možnost implementace jednoduchého příkazového řádku, který by umožnil spouštět příkazy přímo na systému, z prostředí portálu. Pokud by i tato možnost byla pro uživatele omezující, pak SSH umožňuje tunelování připojení a uživatel by pak pomocí SSH připojením k centrálnímu serveru získal přístup i k ostatním zařízením. Při práci s Arduiny může uživatelovi chybět komunikace se zařízením přes sériovou linku. Její implementace by mohla být také součástí rozvoje systému. Práci s Arduiny by také ulehčilo předvyplnění databáze informacemi důležitými pro nahrávání firmwaru do

zařízení pomocí avrdude. Tyto data lze získat z konfiguračních souborů Arduino IDE.

Samotný systém by mohl být zjednodušen sloučením funkcionalit některých skriptů do jednoho. Současný stav je zapříčiněn vývojem různých mechanismů, od kterých nakonec bylo upuštěno, ale zůstala jeho roztříštěnost.

Závěr

Cílem práce bylo navrhnout a vytvořit systém pro programovou správu robotů. Nejprve byly analyzovány podobné systémy a způsob řešení správy robotů a Arduin při výuce na FIT. V literární rešerši byla provedena analýza možností identifikace zařízení, bezpečného připojení k zařízení pomocí SSH, identifikace připojení zařízení pomocí udev a efektivního přenosu dat pomocí rsync.

Na základě zjištěných skutečností pak byl navrhnut a realizován samotný systém. Systém zprostředkovává svoji funkčnost webovému portálu pomocí skriptů. Systém umožňuje spouštění aplikací na zařízení, zálohování zařízení, přenášení souborů na zařízení a další akce. To vše bez nutnosti zadávat heslo uživatelem díky autentizaci pomocí SSH klíčů. Systém umožňuje detekci online zařízení, která poskytuje rychlý přehled o stavu zařízení, zvláště pak v případě robota NAO. Celkem bylo vytvořeno 10 skriptů pro obsluhu zařízení z webového portálu, 3 instalční balíčky pro zařízení a celkově systém obsahuje 24 souborů (skriptů, programů, kofiguračních souborů, ...). Systém obsahuje možnost přidělení jména Arduino zařízení, tato možnost není běžná u podobných systémů.

V budoucnu bude možné systém na základě vypracovaných procesů a skriptů rozšiřovat o nové funkce v závislosti na požadavcích výuky.

Literatura

[1] Microchip Technology Inc.: 24AA02UID [online]. [cit. 2018-04-17]. Dostupné z: http://www.microchip.com/wwwproducts/en/24AA02UID

[2] ARDUINO SOFTWARE: Arduino IDE [software]. Leden 2017, [cit. 2018-04-12]. Dostupné z: https://www.arduino.cc/en/Main/Software

[3] ARDUINO SOFTWARE: Arduino Create [online]. Duben 2016, [cit. 2018-04-12]. Dostupné z: https://create.arduino.cc/

[4] Voda, Z.: ARDUINO CREATE ANEB PROGRAMOVÁNÍ ARDUINA ONLINE [online]. Červenec 2017, [cit. 2018-04-12]. Dostupné z: https://arduino.cz/arduino-create-aneb-programovani-arduina-online/

[5] DATAPLICITY: Dataplicity [online]. Červenec 2017, [cit. 2018-04-13]. Dostupné z: https://www.dataplicity.com

[6] Biggs, J.: Dataplicity lets you access your Raspberry Pi from anywhere [online]. Prosinec 2016, [cit. 2018-04-13]. Dostupné z: https://techcrunch.com/2016/12/13/dataplicity-lets-you-access-your-raspberry-pi-from-anywhere/

[7] Cybergibbons: How can I get a unique ID for all my Arduino boards? [online]. Únor 2014, [cit. 2018-04-17]. Dostupné z: https://arduino.stackexchange.com/a/283

[8] Future Technology Devices International Ltd.: Software Application Development D2XX Programmer's Guide [online]. Únor 2012, [cit. 2018-04-17]. Dostupné z: http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf

[9] Doležel, L.: Píšeme pravidla pro udev [online]. Září 2012, [cit. 2018-04-22]. Dostupné z: http://www.abclinuxu.cz/clanky/piseme-pravidla-pro-udev

[10] Drake, D.: Writing udev rules [online]. Duben 2006, [cit. 2018-04-22]. Dostupné z: http://www.reactivated.net/writing_udev_rules.html

[11] freedesktop.org: udevadm - udev management tool [online]. [cit. 2018-04-23]. Dostupné z: https://www.freedesktop.org/software/systemd/man/udevadm.html

[12] Pillai, S.: Secure Shell: How Does SSH Work [online]. Květen 2013, [cit. 2018-04-24]. Dostupné z: https://www.slashroot.in/secure-shell-how-does-ssh-work

[13] Ladia, A.: How does SSH Work [online]. Listopad 2017, [cit. 2018-04-25]. Dostupné z: https://www.hostinger.com/tutorials/ssh-tutorial-how-does-ssh-work

[14] Red Hat Linux 7.3: The Official Red Hat Linux Reference Guide: Layers of SSH Security [online]. [cit. 2018-04-25]. Dostupné z: https://www-uxsup.csx.cam.ac.uk/pub/doc/redhat/redhat7.3/rhl-rg-en-7.3/s1-ssh-protocol.html

[15] Bernauer, A.: How to get ssh server fingerprint information [online]. Listopad 2007, [cit. 2018-04-25]. Dostupné z: http://www.lysium.de/blog/index.php?/archives/186-How-to-get-ssh-server-fingerprint-information.html

[16] Prikryl, M.: ssh -o PreferredAuthentications: What's the difference between "password" and "keyboard-interactive"? [online]. Březen 2015, [cit. 2018-04-25]. Dostupné z: https://superuser.com/a/894625

[17] SSH.COM: User Authentication with GSSAPI [online]. 2012, [cit. 2018-04-25]. Dostupné z: https://www.ssh.com/manuals/server-admin/62/userauth-gssapi.html

[18] Bárta, M.: Pokročilé zálohování s Rsync [online]. Červenec 2007, [cit. 2018-04-27]. Dostupné z: https://www.root.cz/clanky/pokrocile-zalohovani-s-rsync/

Příloha A

Seznam použitých zkratek

SSH Secure Shell

IDE Integrated development environment

FIT Faculty of Information Technology

IP Internet Protocol

USB Universal Serial Bus

TCP Transmission Control Protocol

BI-ARD Interaktivní aplikace s Arduinem

REST Representational State Transfer

API Application Programming Interface

SQL Structured Query Language

LIVS Laboratoř inteligentních vestavných systémů

MITM Man in the midle

MAC Media Access Control

SCP Secure copy protocol

HTTPS Hypertext Transfer Protocol

EEPROM Electrcally Erasable Read-Only Memory

DNS Domain Name System

NSS Name Service Switch

Příloha B

Uživatelská příručka

B.1 Přihlášení do MySQL databáze klientem pro bash

Systém využívá MySQL klienta a server ve verzi 5.7. Veškeré návody v této části se vztahují k této verzi. Systém není kompatibilní s nižšími verzemi MySQL.

Pro přihlášení k databázi je v systému možno využít dvě varianty přihlášení. První je přihlášení pomocí uživatelského jména a hesla, případně je možné specifikovat IP adresu na, které databáze běží (databáze je z bezpečnostních důvodu přístupná pouze ze serveru, tedy jedinou možností je localhost). Pro uživatelské jméno použijeme přepínač -u pro heslo přepínač -p a pro IP adresu serveru `--host=` (upozornění: příkaz mysql vyžaduje pro přihlášení parametry pro přepínače bez mezery). Příklad takového přihlášení:

```
mysql  -uroot -pheslo --host=localhost
```

Takové přihlášení vyvolá hlášku: „[Warning] Using a password on the command line interface can be insecure.“ Druhý způsob přihlášení je pomocí `--login_path` parametru. Pomocí nástroje mysql_config_editor se vytvoří v souboru mylogin.cnf, který je umístněný v domovském adresáři, bezpečné přihlášení. V tomto souboru jsou přihlašovací údaje šifrovány pomocí AES128. Pro vytvoření přihlášení slouží příkaz:

```
mysql_config_editor set --login-path=login1 --host=localhost
↪  --user=user --password
```

B.2 Udevadm nástroj pro správu udev

Udev také využívá program udevadm (udev management tool) pro zobrazování, monitorování a testování udev pravidel. Zde je uvedeno pár užitečných

příkazu pro obsluhu udeadm. Pro zobrazení informací o zařízení (jejich atributů attr) použijeme:

```
udevadm info -a -n /dev/name_of_device info udev
```

Pro zobrazení informací o zařízení (jejich proměnných env) použijeme:

```
udevadm info -q property -n/dev/name_of_device env info
```

Pro zobrazení zpráv, které dostává udev od jádra systému použijeme:

```
udevadm monitor
```

Pro změně nějakého z pravidel je vhodné provést restart udev (změna by se měla provést ihned pro změně pravidla, ale ne vždy k ní dojde) a spuštění pravidel.

```
udevadm control --reload-rules && udevadm trigger
```

Pro spuštění a správnou funkci udev pravidel na testovacím Raspberry Pi bylo nutné zapnout pro správnou funkci pravidel debuggovací mód. Soubor s informacemi o debuggování nalezneme v /etc/udev/udev.conf. Debuggování zapneme pomocí:

```
udevadm control --log-priority=debug
#následně je nutné ještě provést reset udev pomocí
service udev restart
```

B.3 Generování klíčů

Klíče se generují pomocí příkazu `ssh-keygen -t rsa`. Po jeho zadání budete vyzváni k zadání místa, kam se mají uložit vygenerované klíče. V našem systému jsou klíče uloženy v /etc/ssh-livs-devices/. Dále budete vyzvání k zadání heslové fráze (passphrase), která zabezpečuje soukromý klíč proti odcizení. V současnosti systém zabezpečení hesla pomocí heslové fráze neprovádí, a to z důvodu nutnosti zadávat heslovou frázi po restartu serveru ručně (v budoucnu je plánována instalace agentu, který uchovává heslovou frázi v paměti a zajištění dotazu na heslovou frázi po restartu). Nyní je klíč vygenerován. Společné s klíčem soukromým je vygenerován i klíč veřejný, který se liší v příponě (.pub).

B.4 Použití skriptů

B.4.1 Přidání Arduino zařízení

Přidání zařízení do evidence serveru se provádí pomocí příkazu insert_arduino.sh. Tento příkaz se připojí k zařízení, ke kterému jsou připojená Arduina. Na tomto zařízení vyčte informace o zařízení z udevadm a tyto informace zapíše do databáze. Zapsanými informacemi jsou VENDORID, SERAILID, MANUFACTURE, PRODUCTID a PRODUCTNAME. Tento příkaz má tyto přepínače:

- -n přezdívka pro Arduino zařízení, pod kterou ho lze vyhledat v systému a lze pomocí ní nahrávat do něj aplikace.
- -l název login_path přihlášení do MySQL databáze
- -h nápověda pro tento příkaz
- -p heslo do databáze
- -u uživatelské jméno k databázi
- -t IP adresa databázového serveru
- -i IP adresa zařízení s připojenými Arduiny a uživatelské jméno účtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -k SSH soukromý klíč

Povinnými přepínači jsou: -n pro název zařízení, přihlášení k databázi (buď pomocí login_path nebo uživatelským jménem heslem) a -i IP adresa zařízení s připojenými Arduiny. Ostatní přepínače mění nastavení oproti výchozím konfiguračním hodnotám.

B.4.2 Nahrání firmwaru do Arduino zařízení

Programy pro Arduino v hex formátu se nahrávají pomocí příkazu load_to_arduino_from_server.sh. Tento příkaz přesune nahrávaný soubor do zařízení s připojenými arduiny a vzdáleně spustí program avrdude, který nahraje soubor do zařízení. Systém spouští jednotlivé nahrávání paralelně. Informace důležité pro nahrávání pomocí avrdude si systém načte z databáze. Pokud tyto informace chybí, použije nastavení pomocí dané přepínači -b, -c a -m. Tento příkaz má následující parametry:

- bez přepínače názvy zařízení, do nichž se má program nahrát
- -t zadání hex souboru k nahrání do zařízení
- -h nápověda pro tento příkaz
- -b baud rate se nastaví globálně pro všechna zařízení, do kterých se nahrává

- -m typ mikroprocesoru, který je na Arduino zařízení
- -c programátor, který se použije k nahrávání aplikace do Arduino zařízení,
- -i IP adresa zařízení s připojenými Arduiny a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -p heslo do databáze
- -u uživatelské jméno k databázi
- -T IP adresa databázového serveru
- -k SSH soukromý klíč
- -l název login_path přihlášení do MySQL databáze

Povinnými přepínači jsou: -t pro zadání nahrávaného souboru, přihlášení k databázi (buď pomocí loginy_path nebo uživatelským jménem a heslem) a -i IP adresa zařízení s připojenými Arduiny. Ostatní přepínače mění nastavení oproti výchozím konfiguračním hodnotám, anebo pokud hodnoty nejsou k dispozici (-b, -c a -m). Pro nahrávání aplikací přímo ze zařízení s připojenými arduiny slouží skript load_to_arduino.sh. Na rozdíl od „serverové verze“, nemá přístup k databázi a nemůže tedy využívat doplnění informací pro avrdude. Z toho důvodu umožnuje nahrávání pouze na jedno zařízení. Nastavení je podobné jako u předchozího příkazu (neobsahuje přepínače pro připojení).

B.4.3 Přidání linuxového zařízení

První autentizace zařízení probíhá pomocí skriptu first_login.sh, který pošle zařízení veřejný klíč. Od této chvíle je možné se připojit k zařízení pomocí klíčů. Tento skript je nutné spustit ručně ze serveru, protože vyžaduje, aby uživatel zadal heslo k zařízení. Tento příkaz má následující parametry:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -k SSH veřejný klíč
- -h nápověda pro tento příkaz

Povinným přepínačem je -i pro zadání adresy zařízení. Zařízení se přidá pomocí příkazu insert_robot.sh. Tento skript ověří připojení k zařízení pomocí SSH, přečte mac adresu a jméno zařízení, případně nastaví nové. Poté přidá zařízení do databáze. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -l název login_path přihlášení do MySQL databáze
- -h nápověda pro tento příkaz

- -p heslo do databáze
- -u uživatelské jméno k databázi
- -t IP adresa databázového serveru
- -k SSH soukromý klíč
- -n přezdívka pro zařízení, pod kterou ho lze vyhledat v systému a lze pomocí ní nahrávat do něj aplikace. Pokud není použita, skript se zeptá zařízení na jméno

Povinnými přepínači jsou: přihlášení k databázi (buď pomocí login_path nebo uživatelským jménem a heslem) a -i IP adresa zařízení. Ostatní přepínače mění nastavení oproti výchozím konfiguračním hodnotám

B.4.4 Nahrání programu do zařízení a jeho spuštění

Pro nahrání souboru do zařízení slouží skript s názvem run_on_robot.sh, který pošle zařízení aplikaci ke spuštění a vzdáleně aplikaci spustí. Skript podporuje spouštění C++, C, python kódu a spuštění spustitelné aplikace (např. skript, přeložená binární aplikace, ...). Robot NAO neobsahuje překladače g++ a gcc. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -h nápověda pro tento příkaz
- -k SSH soukromý klíč
- -t zadání souboru k nahrání do zařízení
- -CPP kód se přeloží pomocí g++ se standartem -std=c++11 a spustí se
- -C kód se přeloží pomocí gcc a spustí se
- -P kód se spustí pomocí python interpretu
- -R pouze spustí danou aplikaci

Povinné přepínače jsou: -i pro adresu zařízení, -t pro zadání souboru k nahrání a přepínač pro typ souboru.

B.4.5 Reset zařízení

K provedení restartu slouží skript s názvem restart_device.sh. Tento skript umí také zařízení vypnout. Důležité je zkontrolovat zda uživatel, na kterého jsme přihlášeni, může provádět restart a vypnutí zařízení. Jinak je nutné upravit nastavení. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu

- -h nápověda pro tento příkaz
- -k SSH soukromý klíč
- -D pro vypnutí zařízení
- -R pro restart zařízení

Povinné přepínače jsou -i pro adresu zařízení a -R nebo -D

B.4.6 Záloha zařízení

Záloha zařízení a zároveň i přenášení souborů ze zařízení se provádí pomocí backup_robot.sh. Tento skript přenáší soubory pomocí rsync ze zařízení na server. Tento skript má speciální mód, který slouží k tvorbě záplat. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -h nápověda pro tento příkaz
- -k SSH soukromý klíč
- -c složka pro porovnávání se zdrojovou složkou. Soubory ve zdrojové složce jsou porovnány se soubory ve složce pro srovnání a změny jsou přeneseny do cílové složky. Automaticky zapíná patch mód. Pro porovnávání souborů je doporučeno použít porovnání pomocí kontrolního součtu-o cílová složka.
- -b zdrojová složka na zařízení, pro zálohu celého zařízení použijte „/“
- -s použij kontrolní součet pro porovnávání souborů
- -o výstupní složka na serveru

Povinné přepínače jsou -i pro adresu zařízení, -o pro cílovou složku a -b pro složku zdrojovou.

B.4.7 Aktualizace zařízené

Aktualizace zařízení a zároveň přenášení souborů na zařízení se provádí pomocí příkazu send_to_robot.sh. Tento skript přenáší soubory pomocí rsync ze zařízení na server. Tento skript má speciální mód, který slouží k tvorbě záplat. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -h nápověda pro tento příkaz
- -k SSH soukromý klíč

- -c složka pro porovnávání se zdrojovou složkou. Soubory ve zdrojové složce jsou porovnány se soubory ve složce pro srovnání a změny jsou přeneseny do cílové složky. Automaticky zapíná patch mód. Pro porovnávání souborů je doporučeno použít porovnání pomocí kontrolního součtu -o cílová složka.
- -b zdrojová složka na zařízení, pro zálohu celého zařízení použijte „/“
- -s použij kontrolní součet pro porovnávání souborů
- -o výstupní složka na serveru

Povinné přepínače jsou -i pro adresu zařízení, -o pro cílovou složku a -b pro složku zdrojovou.

B.4.8 Instalace aplikace na zařízení

Instalaci aplikace na robotovi provedeme pomocí skriptu install_app.sh. Tento skript slouží k instalaci pouze na roboty NAO nebo zařízení s linuxovou distribucí Gentoo. Skript pošle instalační soubor robotovi a rozbalí ho v jeho kořenovém adresáři. Tím dojde k instalaci aplikace. Instalační soubor musí mít tar.gz2 formát. Příkaz využívá tyto přepínače:

- -i IP adresa a uživatelské jméno učtu zařízení ve formátu user@ip_address, nebo pouze IP adresa pro použití výchozího uživatelského účtu
- -t tar.gz2 instalační soubor
- -h nápověda pro tento příkaz
- -k SSH soukromý klíč

Povinné přepínače jsou -i pro adresu zařízení a -t pro instalační soubor.

B.5 Seznam souborů

- backup_robot.sh – skript pro zálohu zařízení a posílání souborů ze zařízení.
- check_online_robots.sh – skener který pomocí ping zjišťuje, která zařízení jsou online.
- client_arduino.py – klient pro zařízení s arduiny. Posílá serveru informace o arduinu po připojení/odpojení.
- client.py – klient pro linuxová zařízení. Posílá serveru informace o zařízení po spuštění zařízení.
- client_state.py – klient pro robota NAO. Posílá serveru informace o zařízení po spuštění zařízení.
- device.conf – konfigurační soubor zařízení.

- first_login.sh – skript pro první připojení na zařízení. Zajišťuje výměnu klíčů.
- get_robot_name.py – vrátí jméno zařízení. Jiná implementace pro robota NAO a ostatní zařízení.
- set_robot_name.py – skript nastaví jméno robotovi. Jiná implementace pro robota NAO a ostatní zařízení.
- get_device_type.py – získá typ zařízení.
- insert_arduino.sh – skript pro vložení arduina do databáze a jeho pojmenování.
- insert_robot.sh – skript pro vložení linuxového zařízení do databáze a jeho případné pojmenování.
- install_app.sh – skript který nainstaluje danou aplikaci na robota NAO nebo zařízení s gentoo.
- load_to_arduino_from_server.sh – skript nahraje hex aplikaci do arduino zařízení ze serveru.
- load_to_arduino.sh – skript nahraje hex aplikaci do arduino zařízení ze zařízení s připojenými arduiny.
- plug_in_script.sh – skript registrující do databáze připojení arduino zařízení. Je spouštěn pomocí server.py.
- plug.log – logovací soubor který zaznamenává připojená zařízení k serveru.
- plug_out_script.sh – skript registrující do databáze odpojení arduino zařízení. Je spouštěn pomocí server.py.
- restart_device.sh – skript sloužící k restartování/vypínání zařízení.
- rms_server.conf – konfigurační soubor pro skripty.
- robot_login_sf.sh – skript registrující do databáze připojení linux zařízení. Je spouštěn pomocí server.py.
- run_on_robot.sh – skript spustí na zařízení danou aplikaci nebo přeloží a spustí daný kód (C++, C, python).
- send_to_robot.sh – skript pošle na zařízení daný soubor. Podporuje také obnovení ze zálohy.
- server.conf – konfigurační soubor pro server.py.
- server.py – server přijímající informace o připojení od jednotlivých klientů.

Příloha C

Obsah přiložené SD karty

- `readme.txt` ..... stručný popis obsahu SD karty
- `install_packages` ..... instalační balíčky systému
- `server_files` ..... soubory které jsou v systému umístněné na serveru
- `příručka.pdf` ..... uživatelská příručka systému
- `src`
  - `impl` ..... zdrojové kódy implementace
    - `conf` ..... konfigurační soubory
    - `python` ..... python skripty/programy
    - `bash` ..... bash skripty
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `BP_Kolář_Petr_2018.pdf` ..... text práce ve formátu PDF