

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Monitorovací systém na bázi Raspberry Pi

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2015

Matěj Kareš

Abstract

Monitoring system based on Raspberry Pi

This bachelor thesis deals with the possibilities of using computer Raspberry Pi for security purposes. The thesis is focused on creating monitoring system for small companies, households, garages and/or warehouses. The thesis surveys and analyses operating systems, programming languages for Raspberry Pi and other possible uses of Raspberry Pi for non-security purposes. The thesis contains the process of creating the system including solutions to possible problems.

Abstrakt

Monitorovací systém na bázi Raspberry Pi

Tato práce se zabývá možnostmi použití počítače Raspberry Pi k monitorovacím účelům. Obsah práce je zaměřen na vyhotovení monitorovacího systému pro malé firmy, domácnosti, garáže či sklady. Součástí práce je analýza operačních systémů, programovacích jazyků a jiných možností využití Raspberry Pi mimo oblast monitorování. V textu je zahrnut popis tvorby monitorovacího systému, včetně možných problémů.

Obsah

1	Úvod	1
2	Raspberry Pi	2
2.1	Obecné využití Raspberry Pi	4
2.2	Využití Raspberry Pi k zabezpečovacím účelům	4
2.2.1	Monitorovací systém pomocí programu Motion	5
2.3	Operační systémy pro Raspberry Pi.....	6
2.3.1	Arch Linux.....	7
2.3.2	Android 2.3.....	8
2.3.3	Plan 9	9
2.3.4	Risc OS.....	10
2.3.5	Raspbian.....	11
2.3.6	Volba vhodného operačního systému.....	11
2.4	Výběr programovacího jazyka.....	12
3	Analýza problému	13
3.1	Modelová situace	13
3.2	Knihovny pro ovládání kamery.....	13
3.3	Analýza klientské části	14
3.4	Analýza serverové části.....	15
4	Realizace monitorovacího systému	15
4.1	USB webkamera a kamerový modul	16
4.2	Ochrana a umístění Raspberry Pi.....	16
4.3	Ukládání snímků	17
4.4	Rozpoznání pohybu pomocí kamery.....	17
4.5	Využití V/V pinů.....	18
4.6	Obecný model klientské části.....	18
4.7	Obecný model serverové části.....	19

4.8	Web	20
5	Předpokládané a dosažené výsledky	21
5.1	Snímkovací frekvence a rozlišení	21
5.2	Nároky na paměť a procesor.....	22
5.3	Cena a efektivita	23
6	Zhodnocení práce a získané zkušenosti.....	24
7	Doporučení pro další vývoj.....	25
7.1	Doporučení s vysokou prioritou	25
7.2	Doporučení s nízkou prioritou	25
8	Závěr	26
	Reference.....	27
	Příloha - Uživatelská příručka.....	29
	Překlad aplikace	29
	Instalace a konfigurace.....	29
	Spuštění	30
	Ovládání.....	30

1 Úvod

Krádeže a podvody jsou s lidstvem spojeny už od vzniku prvních civilizací a lidé stále hledají, jak trestnou činnost odhalit, prokázat a potrestat. Mnohem lepší je ale trestné činnosti předejít například vystrašením kriminálního. Obchodníci často nakoupí zboží, které se stále nachází v objektu i v nepřítomnosti majitele či zaměstnanců. Takto nechráněné zboží láká zloděje a stresuje samotné podnikatele, proto je významnou součástí takovýchto obchodů zabezpečovací systém, ať už se jedná pouze o kameru k uchování evidence případného zločinu, nebo o alarm napojený na policejní stanici, který může nejen přivolat policii, ale i lupiče vystrašit a tak předejít dalším škodám na majetku. Tento problém se však netýká pouze obchodů a malých firem, ale i objektů jako jsou domácnosti, garáže, dílny atd.

Cílem práce je prozkoumat možnosti využití počítače Raspberry Pi pro monitorovací účely a vytvořit jednoduchou aplikaci, která využití demonstruje. Předpokládá se, že Raspberry Pi bude napájeno přímo z elektrického rozvodu v monitorovaném objektu a bude mít přístup do sítě internet. Cílovými skupinami, které mohou uplatnit popsané poznatky nebo přímo využít vytvořenou aplikaci, jsou malé firmy, obchody, domácnosti, garáže a pokud by k navrženému systému byla připojena baterie a v jeho dosahu Wi-Fi signál, tak by zařízení mohlo být využito i např. myslivci a rybáři.

2 Raspberry Pi

Raspberry Pi je jednodeskový počítač o velikosti platební karty (obrázek 1). Vyvíjí ho britská nadace *Raspberry Pi Foundation*¹, jejímž cílem je podpořit výuku informatiky ve školství.

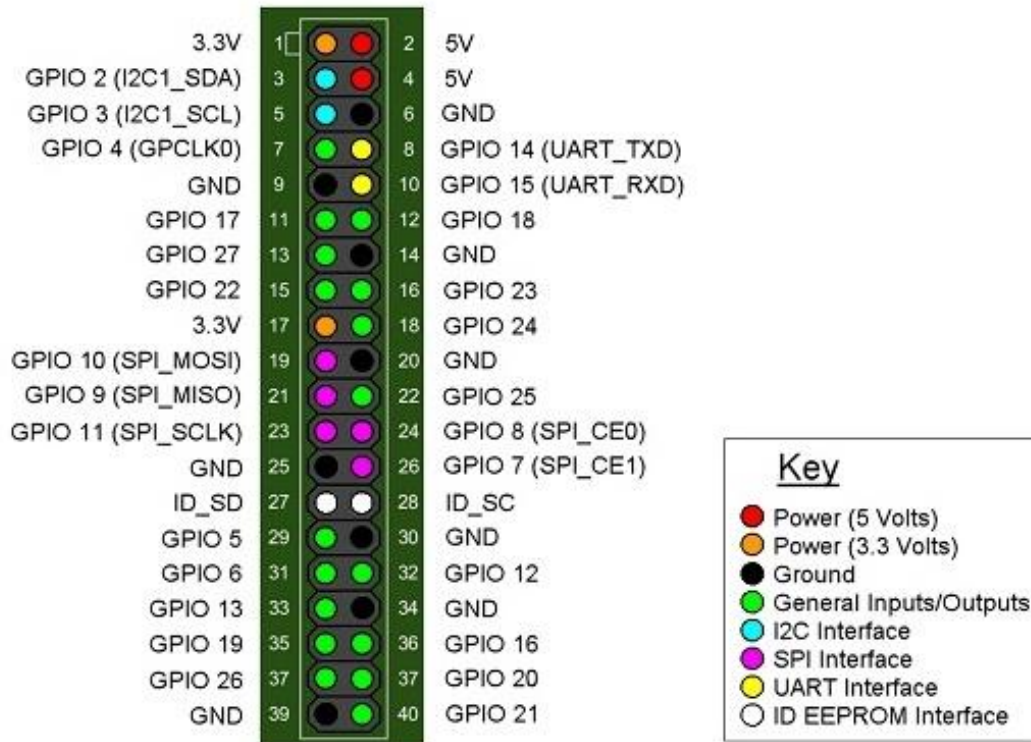


Obrázek 1 - Raspberry Pi model B+

Základem Raspberry Pi je jednočipový počítač *BCM2835* firmy *Broadcom*, který obsahuje centrální procesor *ARM1176JZF-S* na taktu 700MHz, grafický procesor *VideoCore IV*, obrazový výstup s rozhraním HDMI a zvukový výstup přes 3,5mm konektor nebo prostřednictvím HDMI. Od svého vzniku bylo RPi vylepšováno a dnes již existuje několik jeho modelů. Prvním modelem byl model A, který měl, krom již popsaných částí, 256 MB RAM, slot pro SD nebo MMC kartu, jeden USB port a 26 vstupně-výstupních (dále jen V/V) pinů. Nástupcem modelu A je model A+, který jen nahradil slot na SD karty slotem na mikro SD karty. Dále existuje model B, jehož operační paměť byla oproti modelu A zdvojnásobena na 512 MB,

¹ <https://www.raspberrypi.org/>

počet USB portů se zvýšil na 2 a nově přibyl ethernetový adaptér 10/100 s konektorem RJ56. Model B byl později rozšířen na model B+, který má celkem 4 USB porty. Na modelech A+ a B+ se také nachází 40 V/V pinů (obrázek 2) pro připojení např. snímačů, LED, piezo atd. Raspberry Pi vyniká nízkou spotřebou, která nepřesáhne 3W.



Obrázek 2 – Model A+ a B+ V/V Piny²

V současné době existuje Raspberry Pi 2 se čtyřjádrovým procesorem *BCM2836 ARM Cortex-A7* na taktu 900MHz, dvoujádrovým GPU *VideoCore IV* a 1GB RAM. Bohužel Raspberry Pi 2 nebylo dostupné v době zahájení projektu a tak bude pro návrh monitorovacího systému využít model Raspberry Pi (1) B+.

V následujících kapitolách se podíváme na obecné využití Raspberry Pi, následně se zaměříme na oblast zabezpečení pomocí Raspberry Pi a probereme dostupné operační systémy a programovací jazyky.

² Obrázek přejatý z <http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus>

2.1 Obecné využití Raspberry Pi

Raspberry Pi je plnohodnotný počítač a lze ho tak využít k obvyklé každodenní práci na počítači jako například prohlížení e-mailů, webových stránek, sledování videa, tvorbě dokumentů, programování atd. Díky svým rozměrům a spotřebě se stává kvalitní náhradou za slabší domácí počítače. Na Raspberry Pi se nenachází žádný ventilátor a tak během provozu nevydává zvuk, čímž umožňuje tichou práci. Existuje mnoho rychlých průvodců možnostmi Raspberry Pi, např. [1, 2, 3, 4].

Velmi častým využitím Raspberry Pi jsou také síťová datová úložiště (dále jen NAS – *Network Attached Storage*)³. Programem Samba⁴ lze vytvořit z Raspberry Pi síťový disk, aby měla zařízení ve stejné síti přístup ke společným datům. Pro tyto účely je ale potřeba vybavit Raspberry Pi například flash diskem nebo externím diskem, jinak je velikost a rychlost úložiště limitována použitou SD kartou s operačním systémem. Po nainstalování Apache⁵, nebo podobného programu, lze z Raspberry Pi udělat webserver⁶ pro např. osobní stránky, e-shop atd.

Malé rozměry a spotřeba také umožňují umístění Raspberry Pi s baterií na místa jako skleníky, posedy, studny nebo jízdní kola ke sběru různých typů dat⁷. Raspberry Pi je vybaveno 40 V/V piny pro připojení snímačů nebo jiných elektronických zařízení. Tyto piny dobře slouží i k ovládní motorků a ramen a tak se dají pomocí Raspberry Pi vytvořit i roboti a podobné automaty. Raspberry Pi se také využívá k automatizaci domácností – regulace teploty, ovládní světel, otevírání dveří garáže atd. Více o automatizaci domácností lze najít např. v [5]. Různé projekty založené na Raspberry Pi lze také nalézt např. v [6].

2.2 Využití Raspberry Pi k zabezpečovacím účelům

Jak bylo již v úvodu nastíněno, existuje více způsobů možné ochrany uvedených objektů. Alarm je první věc, která se nám vybaví po vyřčení slova zabezpečení, velkou výhodou alarmu je možnost zastrášení lupiče, avšak ke spuštění alarmu je potřeba nějaký senzor např. senzor pohybu či kamera. Kamerový systém může

³ <http://www.howtogeek.com/139433/how-to-turn-a-raspberry-pi-into-a-low-power-network-storage-device/>

⁴ <https://www.samba.org/>

⁵ <https://www.apache.org/>

⁶ <http://readwrite.com/2014/06/27/raspberry-pi-web-server-website-hosting>

⁷ <http://www.instructables.com/id/Raspberry-Pi-Completely-Wireless-IP-Camera-Solar/>

sloužit k monitorování chráněné oblasti a evidenci případného zločinu. Senzory pohybu jsou ideální pro prostory s nedostatkem světla. Příkladem takového senzoru může být infračervená brána – je-li přerušena paprsek mezi dvěma body brány, tak se vyvolá akce. Více o zabezpečení domácnosti zde [7] a další informace o monitorování zde [8].

K Raspberry Pi lze prostřednictvím USB portu připojit libovolnou webkameru a jen výjimečně se stane, že je kamera nekompatibilní. Stejným způsobem lze zařízení vybavit i mikrofonom nebo lze rovnou využít webkamery s vestavěným mikrofonom, pokud je pro monitorování důležitý i zvuk. Alternativou pro USB webkameru je kamerový modul⁸ nabízený výrobcem Raspberry Pi, který se cenově pohybuje na úrovni levnějších USB webkamer, ale zvládá větší rozlišení i snímkovací frekvence. Prostřednictvím V/V pinů lze připojit např. alarm, spínač osvětlení nebo jiné periférie sloužící k zastrašení nebo signalizaci.

K Raspberry Pi je možné přes USB rozhraní připojit Wi-Fi modul a tak se zbavíme ethernetového kabelu. Raspberry Pi má tedy rozsáhlou paletu možností připojení periferních zařízení, které mohou sloužit k zabezpečovacím účelům. Napájení je možné realizovat pomocí baterií, avšak to výrobce nedoporučuje, protože při nesprávné manipulaci je možné zařízení poškodit.

Raspberry Pi je tedy hardwarově dostatečně vybavené pro účely zabezpečení menších objektů. Volně dostupné softwarové vybavení pro monitorovací účely může být například program Motion (viz následující kapitola). Cenově se tedy pohybujeme na ceně Raspberry Pi (cca 700 Kč – v závislosti na modelu) a kamery (cca 500 Kč), což vychází na přibližně 1200 Kč, kdežto konkurenční produkty stojí více než 3000 Kč⁹. Jelikož nevytváříme žádnou nedostupnou technologii, bude hlavní motivací práce cena a efektivita.

2.2.1 Monitorovací systém pomocí programu Motion

Motion¹⁰ je software pro monitorovací účely, který dokáže rozpoznat významnou změnu v obraze, jinak řečeno rozpozná pohyb ve snímaném obraze. Tento software

⁸ <https://www.raspberrypi.org/products/camera-module/>

⁹ http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHometechnet.idnes.cz/test-mala-ceska-gsm-kamera-vam-pohlida-dum-chatu-ci-garaz-pjk-/tec_foto.aspx?c=A101115_123032_tec_foto_vok

¹⁰ <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>

je vyvíjen pro Linux pod open-source licencí. Jedná se o konzolovou aplikaci, která snímá obraz z USB webkamery, případně z IP kamery a tento obraz vysílá na vlastním webservru. Pokud zadáte z jiného počítače IP adresu počítače, kde je program Motion spuštěn, a zadáte port, na který je vysílání nastaveno, uvidíte aktuální obraz z kamery. Motion dokáže rozpoznávat pohyb a případně zvyšovat snímkovací frekvenci pro lepší záznam pohybu. Problém ale nastává ve chvíli, kdy případný zloděj zařízení ukradne, v tento moment ztrácíme živý přenos i veškerá nasnímaná data a tedy nemáme jak zločin později prokázat. Řešením by bylo upravení programu tak, aby data ukládal na jiný počítač, ke kterému zloděj nemá přístup.

2.3 Operační systémy pro Raspberry Pi

Raspberry Pi nám lépe poslouží, pokud na něm bude nějaký operační systém (dále jen OS). Kvůli struktuře naší aplikace hledáme OS, který je kompaktní a rychlý. Drtivá většina OS na Raspberry Pi je na bázi Linuxu a jeho distribucí a tak nezkušeným uživatelům doporučuji alespoň základní seznámení s tímto systémem, viz [9]. Uživatelské rozhraní můžeme do jisté míry zanedbat, jelikož hlavním účelem tohoto projektu je zaznamenávat data z kamery a popřípadě v nich rozpoznávat určité jevy. Předpokládá se, že hlavním přehrávacím médiem nebude samotné Raspberry Pi, ale vzdálený počítač, na který nasnímaná data ukládáme. Co se týká OS dostupných pro Raspberry Pi, tak jsem se zaměřil na pět, mezi majiteli Raspberry Pi, nejpopulárnějších OS:

1. Raspbian¹¹
2. Risc OS¹²
3. Plan 9¹³
4. Android¹⁴
5. Arch Linux¹⁵

Existuje ještě několik dalších OS, avšak ty se mnoho neliší od výše zmíněných a funkčností jsou stejné nebo slabší. Jejich výhoda může spočívat například

¹¹ <http://www.raspbian.org/>

¹² <https://www.riscosopen.org/content/>

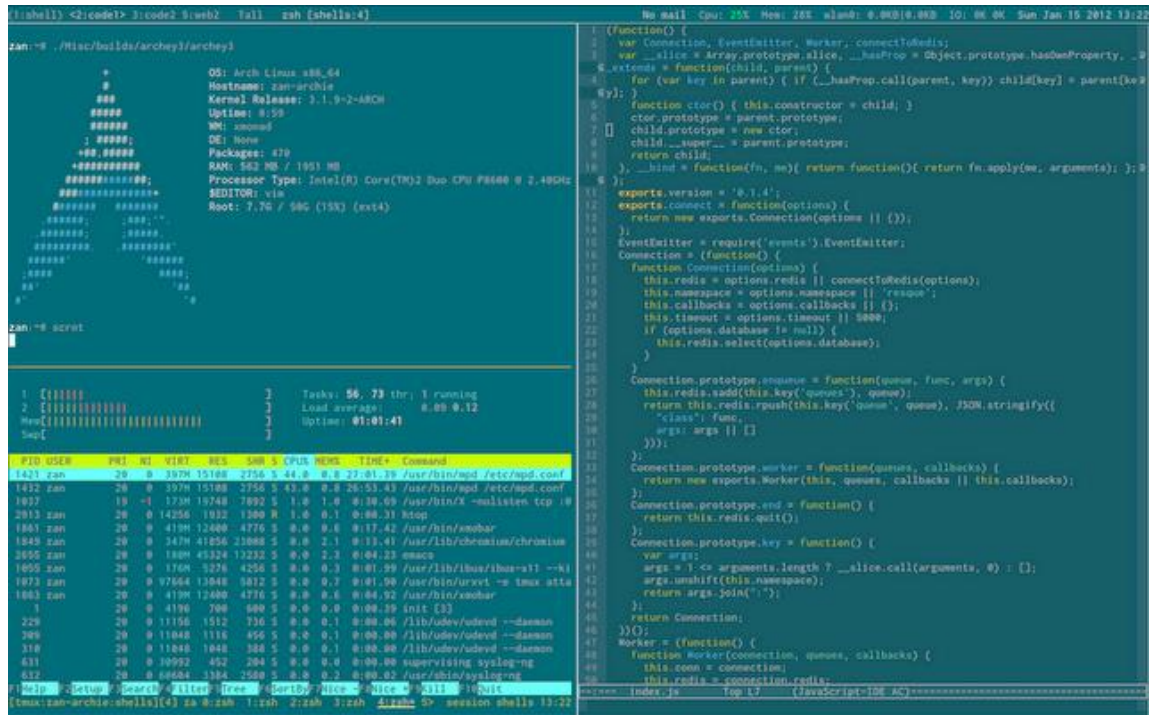
¹³ <http://plan9.bell-labs.com/plan9/>

¹⁴ <http://developer.android.com/about/versions/android-2.3.3.html>

¹⁵ <https://www.archlinux.org/>

v příjemnějším uživatelském rozhraní nebo v bohatší sadě předinstalovaných aplikací a her, což pro naši aplikaci není atraktivní.

2.3.1 Arch Linux

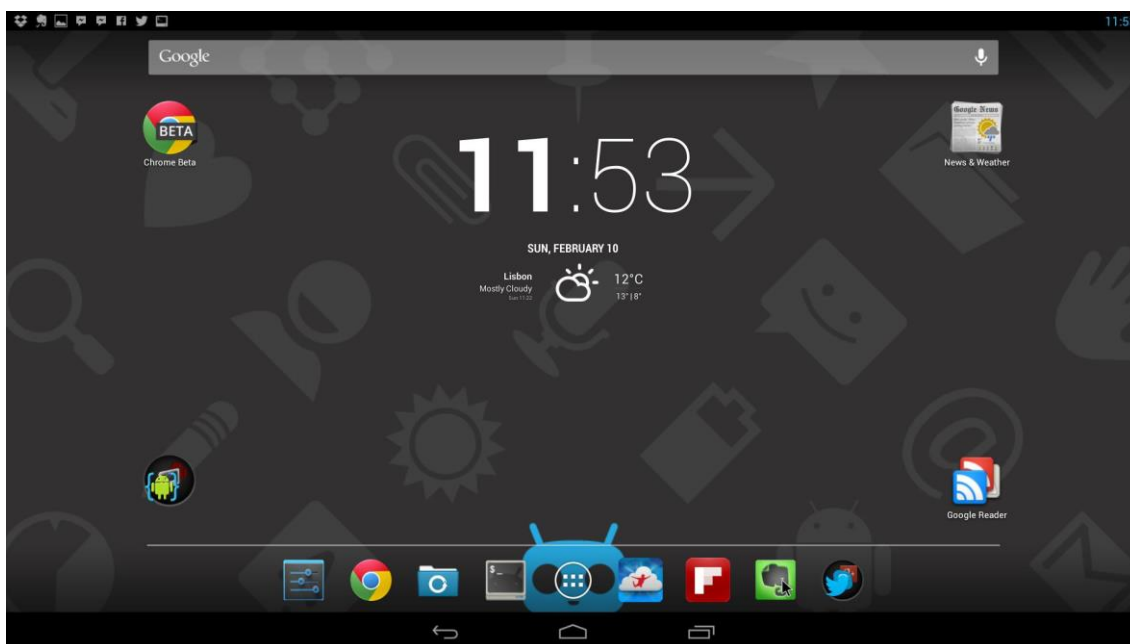


Obrázek 3 - Náhled na Arch Linux

Po prvním spuštění Archu se ocitáme v terminálovém okně a je jen na uživateli, aby si stáhl, nainstaloval a nakonfiguroval jednotlivé komponenty systému sám. Ačkoliv se toto může zdát jako přítěž, je takto instalovaný OS přesně určen potřebám uživatele, neobsahuje zbytečné funkce a je svižnější, než jeho konkurenti. Mnoho uživatelů však dá přednost snadněji nastavitelným alternativám za cenu větší velikosti a menšího výkonu. Samozřejmě je možné si doinstalovat grafické uživatelské rozhraní (dále jen GUI), jako například OpenBox¹⁶ (obrázek 3), ale opět musím zmínit, že i přesto, že by tak Arch byl z naší pětky OS nejlepší, je konfigurace příliš složitá a časově náročná, aby si získal oblibu u uživatelů. Přehrávání médií je opět možné pouze po dodatečné konfiguraci. Celkově bych tedy Arch hodnotil jako složitou stavebnici, ze které může vzniknout dobrý OS, ale vložená námaha se vyplatí pouze, pokud výkon konkurentů není dostatečný.

¹⁶ http://openbox.org/wiki/Main_Page

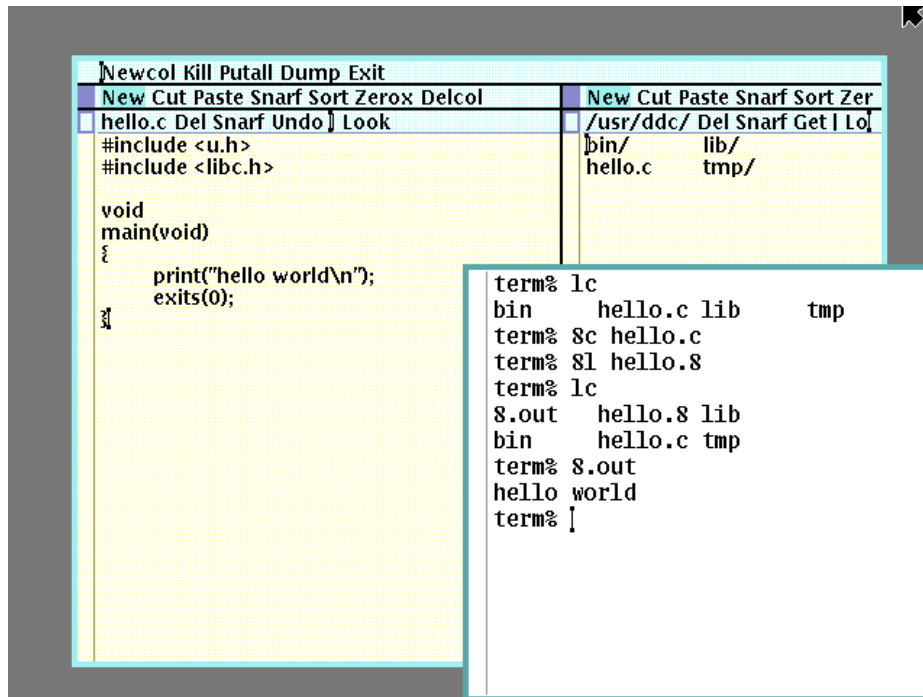
2.3.2 Android 2.3



Obrázek 4 – Screenshot z OS Android 2.3 na Raspberry Pi

Android nabízí velmi příjemné uživatelské rozhraní (obrázek 4), které na Raspberry Pi chybí, poměrně často padá a je velmi pomalé. Naštěstí pro nás používáme 512MB RAM verzi Raspberry Pi, která dokáže být pro potřeby Androidu dostatečná. Základní softwarové vybavení je velkým plusem Androidu - nechybí zde přehrávač hudby a videa, ani webový prohlížeč. Bohužel je ale již tolik paměti zabráno samotným OS, že přehrávání médií je bojem o paměť mezi OS a přehrávačem (videa se často zasekávala a občas vznikaly obrazové fragmenty). Celkově bych Android hodnotil jako příjemně vypadající OS s připravenou základní sadou software, avšak jeho objemnost ho pro Raspberry Pi model B+ činí omezeně použitelným až nepoužitelným kvůli nedostatečně velké operační paměti a sám výrobce Raspberry Pi ho nedoporučuje a označuje pouze za experiment.

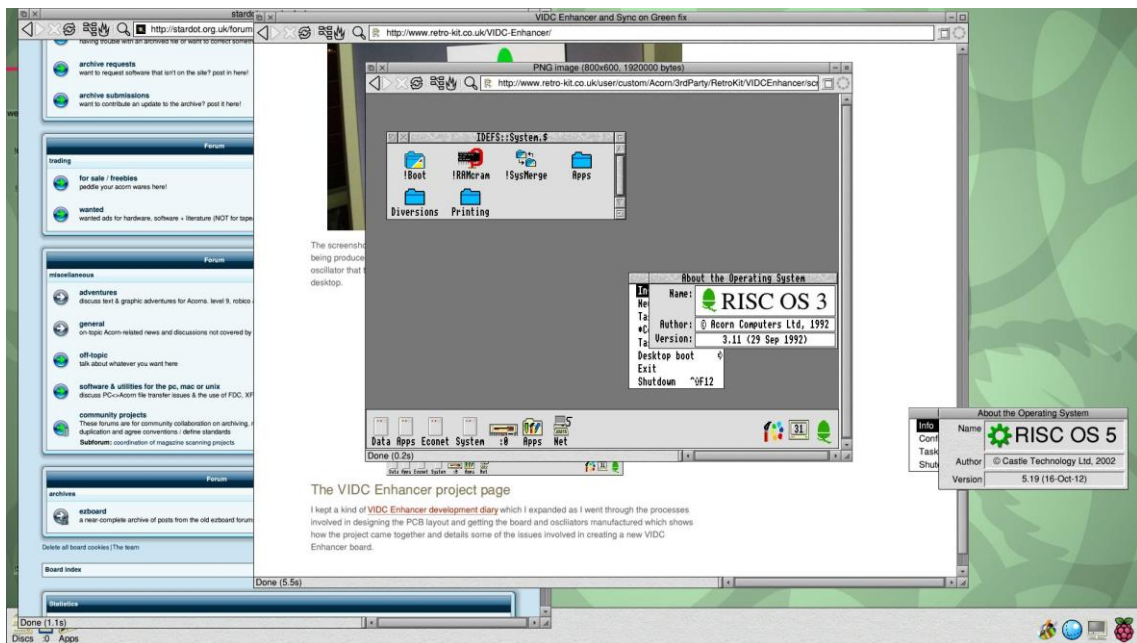
2.3.3 Plan 9



Obrázek 5 - Screenshot z Plan 9 OS

Uživatelské rozhraní (obrázek 5) je velmi odlišné od rozhraní, na která jsme zvyklí. Příkazy systému jsou téměř totožné s příkazy Unixu a jsou navíc obohacené o alternativní zápisy (aliases). Na internetu existuje velmi dobrá dokumentace. Celkově grafické rozhraní neobsahuje žádné nadbytečnosti a je určeno pro vývojáře a nadšence. Okna nemají dekorace, plocha je jen šedivá obrazovka. Učení se se systémem není ale tak intuitivní a proto je velmi zdlouhavé. Celkově bych Plan 9 hodnotil jako dostačující OS z hlediska výkonu, ale složitá práce s rozhráním ho posouvá do stínu následujících OS.

2.3.4 Risc OS



Obrázek 6 - Screenshot z Risc OS na Raspberry Pi

Uživatelské rozhraní je již na první pohled mnohonásobně příjemnější, než u předchozích OS - barevné, dobře rozložené, s „retro“ nádechem (viz obrázek 6). Risc OS podporuje rozlišení 1080p, což je také příjemná změna. Základní softwarové vybavení, jako např. webový prohlížeč, souborový manažer nebo kalkulačka, jsou ihned dostupné a k dispozici je i balíčkový manažer, kde si můžeme doinstalovat potřebné aplikace. Ovládání systému je intuitivní a informace o systému jsou dostupné na internetu. Ale narazil jsem i na nevšední věci, např. síťová karta je v základním režimu vypnutá a zapnout jí (i s návodem) dalo mnoho práce. Celkově hodnotím Risc OS jako velmi pokročilý oproti předchozím – je zde určitá rovnováha mezi hezky vypadajícím a rychlým OS.

2.4 Výběr programovacího jazyka

Na Raspbianu jsou implicitně dostupné jazyky Python, Java, C, C++, Scratch a Ruby. Výsledná aplikace bude monitorovací/zabezpečovací systém. Bude tedy potřeba pracovat s periferními zařízeními (např. webkamera, senzory, alarm) a videem. Aplikace bude obsahovat GUI a nasnímaná data bude ukládat lokálně a posílat do internetu.

Vhodnými jazyky tedy jsou C/C++, Python a Java. Pro příznivce programovacího jazyka Python existuje velmi dobře zpracovaná kniha [10]. Nejlepší variantou je jazyk Java, který je multiplatformní, vysokoúrovňový a poskytuje dostatečné nástroje pro komunikaci po internetu a pro vytváření GUI. Nevýhody Javy jsou její velikost a rychlost. Java nás navíc šetří rutinních postupů, jako např. alokace paměti. Bohužel ne vždy potřebujeme vše, co za nás Java dělá, proto občas přijdeme o mnoho paměti nebo výkonu procesoru zbytečně. Více o programovacím jazyku Java viz [11] a [12].

Raspberry Pi model B+ disponuje 512MB RAM, což se může zhýčkanému uživateli zdát málo, avšak naše aplikace bude mít i tak dostatečnou rezervu pro svůj běh. Takt procesoru je 700MHz, ale naše aplikace není reálnová hra a menší takt nám nevadí, naopak může být plusem, protože nám Raspberry Pi vydrží déle na případné baterie. Tyto argumenty předkládám k obhajobě volby jazyka Java pro svou aplikaci.

3 Analýza problému

Cílem této kapitoly je analyzovat problém a navrhnout monitorovací systém, jehož prototyp bude výstupem této práce. Během analýzy budou popsány i alternativní přístupy, aby práce pokryla širší možnosti uplatnění.

3.1 Modelová situace

Modelovou situací, ze které budeme pro představu vycházet, je monitorování garáže – malá místnost obsahující cenné předměty (auto, nářadí, popř. stroje). Monitorování bude probíhat pomocí webkamery připojené k Raspberry Pi přímo v objektu a snímaná data se budou odesílat na vzdálený počítač, kde budou zobrazena připraveným webservrem. Místnost bude v době monitorování osvětlena a bude dostupné připojení k internetu nebo lokální síti, ať už prostřednictvím Wi-Fi nebo ethernetovým kabelem. Zde chci zdůraznit, že připojení není podmínkou, ale bez komunikace s vnějším světem nemůžeme kontaktovat majitele objektu a data se budou ukládat pouze lokálně, což může při krádeži Raspberry Pi vést i k případné ztrátě dat. Řešením by bylo uchovávat Raspberry Pi na nenápadném či nedostupném místě nebo použít GSM modul¹⁷ pro jeho lokalizaci v případě odcizení.

3.2 Knihovny pro ovládání kamery

Pro práci s kamerou můžeme využít množství knihoven vytvořených právě pro programovací jazyk Java. První takovou knihovnou je JMF (*Java Media Framework*)¹⁸. JMF poskytuje API (*Application Programming Interface*) pro práci s videem a zvukem. Bohužel nefunguje nativně na Raspberry Pi, jelikož je jeho součástí knihovna *libcivil.so* zkompileovaná na X86 architektuře¹⁹ a pro správnou funkci by bylo potřeba ji překompilovat na Raspberry Pi.

Knihovna *JavaCV*²⁰ vznikla jako Javovská verze knihovny *OpenCV* a poskytuje rozhraní k projektům, které se týkají počítačového vidění²¹. Bohužel k instalaci

¹⁷ GSM (Globální Systém pro Mobilní komunikaci) modul umožňuje odeslání SMS

¹⁸ <http://www.oracle.com/technetwork/articles/javase/index-jsp-140239.html>

¹⁹ <https://www.raspberrypi.org/forums/viewtopic.php?f=34&t=18682>

²⁰ <https://github.com/bytedeco/javacv>

²¹ Počítačové vidění je obor umělé inteligence a rozpoznávání ve výpočetní technice, který se zabývá získáváním informací z nasnímaného obrazu.

vyžaduje stažení balíčků o velikosti přesahující 140MB, chybí k ní *JavaDoc*²² a knihovna není přenositelná. Dalšími možnými knihovnami jsou LTI-CIVIL a FMJ, ale ty jsou zastaralé a jejich tvůrci je dále nevyvíjí ani neudržují.

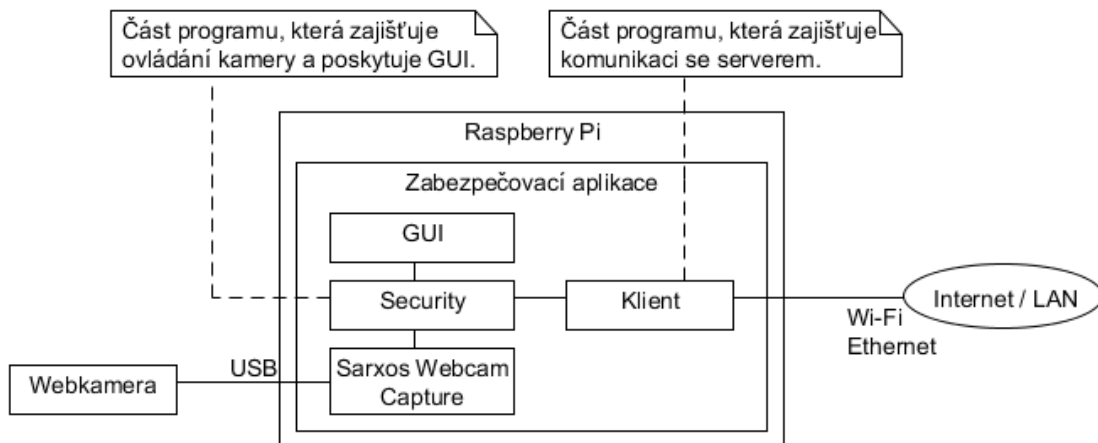
Knihovna *Sarxos Webcam Capture* [13] poskytuje bohaté API k vestavěné nebo externí webkameře. Je navržena tak, aby zpřístupnila nejčastěji využívané funkce webkamer (pořizování snímků, detekce pohybu atd.) a podporuje více zaznamenávacích frameworků²³ (viz [13]). Tato knihovna přesně vyhovuje našim požadavkům. Autor se stará o pravidelné aktualizace, poskytuje okamžitou podporu v případě komplikací a navíc je knihovna vydávána pod open-source licenci. Tyto argumenty předkládám k volbě knihovny *Sarxos Webcam Capture*.

3.3 Analýza klientské části

Jak bylo již zmíněno v části 3.1, projekt se bude skládat ze dvou částí – Raspberry Pi (dále také označováno jako klient) a server. Jádrem aplikace bude část *Security*, která se bude starat o chod programu, úvodní nastavení aplikace a bude poskytovat informační výstup v případě, kdy uživatel bude chtít aplikaci používat jen v textovém režimu. Další částí aplikace bude GUI pro lokální zobrazení přenosu (na monitoru, je-li dostupný), které umožní umístění ovládací aplikace (např. uložit snímek, prohlížet zaznamenané události atd.) a zobrazení stavových informací (snímkovací frekvence, rozlišení, stav spojení se serverem, zaznamenané události apod.). Komunikace s kamerou bude zprostředkována knihovnou *Sarxos Webcam Capture* a komunikace se serverem bude navazována a ukončována částí *Klient*. Na obrázku 8 vidíme diagram nasazení Raspberry Pi s monitorovací aplikací.

²² Dokumentační standard programovacího jazyka Java, případně samotná dokumentace.

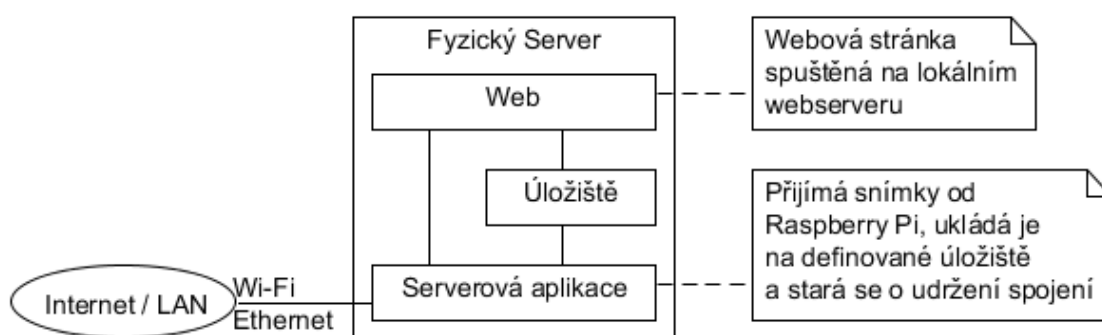
²³ Framework je softwarová vrstva starající se o usnadnění vývoje převzetím typických problémů dané oblasti na sebe, aby se programátor mohl soustředit pouze na své zadání.



Obrázek 8 – Diagram nasazení – Raspberry Pi (klient)

3.4 Analýza serverové části

V internetu bude server. Ten bude data uchovávat na svém disku a zobrazovat prostřednictvím webového rozhraní. To také umožní prohlížení v historii zaznamenaných událostí. Serverová aplikace i web budou využívat společné úložiště, kam server bude ukládat přijaté snímky a web je bude zobrazovat. Součástí serveru bude konfigurační průvodce, jehož účelem bude nastavit společné úložiště pro ukládání snímků a následné zobrazování na webu. Alternativním přístupem by byla databáze, avšak práce se soubory bude na velmi primitivní úrovni a databáze by byla nadbytečná. Zjednodušený diagram nasazení si můžeme prohlédnout na obrázku 9.



Obrázek 9 – Diagram nasazení – Server

4 Realizace monitorovacího systému

V této kapitole si popíšeme klíčové prvky aplikace, jejich implementaci a diskutujeme vhodnost zvolených technologií včetně jejich alternativ.

4.1 USB webkamera a kamerový modul

Nyní je potřeba vybrat zařízení pro záznam. V kapitole 2.2 jsem již zmínil existenci kamerového modulu pro Raspberry Pi (na obrázku 10 vpravo) a vyzdvihoval jsem jeho kvality (rychlost a rozlišení) oproti běžným USB webkamerám (na obrázku 10 vlevo). Bohužel tento modul je zaměřený pouze na Raspberry Pi a nemá využití bez něj. USB webkamery jsou snadno připojitelné díky USB rozhraní a je možné je snadno nahradit v případě poškození. USB webkamery jsou také dostupnější pro nákup a právě díky široké škále výběru kamer může uživatel hýbat s cenou výsledného systému. Tyto skutečnosti mě vedou k používání USB webkamery a podle toho se bude ubírat i směr vývoje aplikace.



Obrázek 10 - USB webkamera a kamerový modul

4.2 Ochrana a umístění Raspberry Pi

K Raspberry Pi existuje množství krabiček, do kterých lze Raspberry Pi umístit a upevnit pomocí 4 předvrtaných děr v základní destičce Raspberry Pi. Cenově se takové krabičky pohybují kolem 200 Kč. Někteří uživatelé si ale raději sami vyrábějí vlastní krabičky a někteří nechávají Raspberry Pi bez jakéhokoliv krytí. S přehříváním nemá Raspberry Pi díky své nízké spotřebě žádný problém a tak případná krabička s několika otvory nepovede k přehřátí. Krabičky lze později podle typu přivrtat na zeď nebo jen přilepit na zvolené místo.

Umístění záleží na konkrétní situaci a ne vždy je možné vše úspěšně skrýt, zvláště kvůli připojeným kabelům (napájení, ethernet). V této práci počítám s možností odcizení Raspberry Pi, proto, dojde-li k odcizení samotného zařízení, data budou uložena na specifikovaném serveru.

4.3 Ukládání snímků

Během monitorování vznikne mnoho snímků, což by mohlo v dlouhodobém měřítku vytvořit problém s nedostatkem místa na disku²⁴. Je tedy potřeba filtrovat nadbytečná data. Cílem monitorování je získat snímky, na kterých se něco děje a kdy daná událost nastala. Naopak nás nezajímají snímky, na kterých se nic neděje. Jednoduchým řešením je ukládat pouze zmíněné zajímavé snímky a ostatní zahazovat. Výjimkou bude snímek ve webovém rozhraní, který bude odeslán a uložen na požádání uživatele.

Zmíněný čas události můžeme vykreslit přímo do snímku, nebo snímky pojmenovávat podle času pořízení. Vykreslování do snímku nám zajišťuje bezpečnější uchování času, ale je výpočetně náročnější, proto jsem zvolil zapisování času do názvu souboru. To nám také umožní lepší orientaci a vyhledávání v nasnímaných datech. Jinou možností by bylo využití obou způsobů zápisu času.

4.4 Rozpoznání pohybu pomocí kamery

Rozpoznávání pohybu, tedy detekce významné změny části snímané scény je přímo součástí knihovny *Sarxos Webcam Capture*. V dokumentaci²⁵ se dozvídáme o proměnné *pixelThreshold*. Tato proměnná může nabývat hodnot 0 až 255 a její hodnota udává, o kolik se musí změnit složky RGB modelu, aby došlo k detekci změny obrazu. Nespecifikujeme-li jinak, je implicitní hodnota nastavena na 25. Po několika rychlých experimentech jsem dospěl k závěru, že implicitní nastavení funguje spolehlivě na denním světle i v temné místnosti, kde jsem byl osvětlen pouze monitorem. S tímto nastavením nedocházelo k falešným vyhodnocením a při pokusu o pomalý pohyb před kamerou jsem byl správně vyhodnocen jako pohybující se objekt.

²⁴ U Raspberry Pi se jedná o použitou SD kartu, která je brána jako disk.

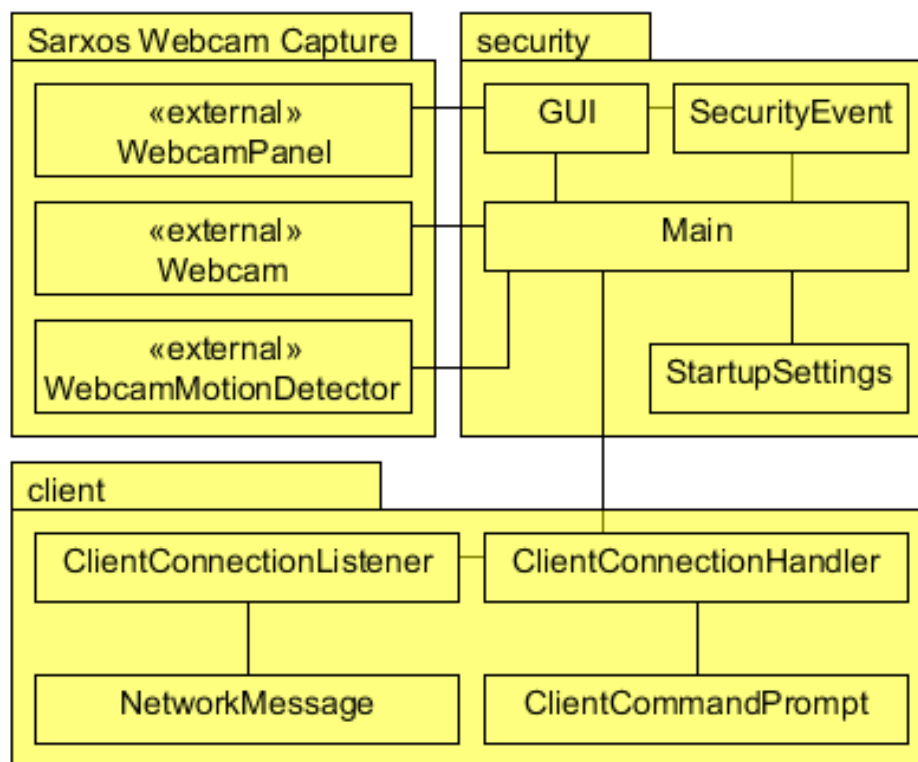
²⁵ <http://webcam-capture.sarxos.pl/site/webcam-capture/apidocs/index.html?com/github/sarxos/webcam/WebcamMotionDetector.html>

4.5 Využití V/V pinů

Poslední pin (pin 40, pro rozložení pinu viz obrázek 2) je nastavený jako vstupní a při změně stavu (z logické 0 na logickou 1 a naopak) odešle zprávu na server. Takto připravený pin může sloužit například ke hlídání oken a dveří. Jednoduše uzavřeme obvod mezi napájecím pinem (pin 1) a zmíněným vstupním pinem pomocí vodiče. Tento vodič přerušíme v místě styku dveří či okna tak, aby se konce dotýkali, je-li průchod uzavřen a rozpojili po otevření. Toto je nejjednodušší způsob využití. Ke vstupnímu pinu lze připojit i jiná snímací zařízení, jejichž výstup je logická 0 (0 Volt) nebo logická 1 (3,3 Volt). Je dobré mít na paměti, že vyšší napětí než 3,3 Volt může pin nevratně poškodit. K realizaci této části projektu jsem využil knihovnu *Pi4J* [14]. Knihovna nabízí velmi jednoduché rozhraní pro ovládání V/V pinů, je kvalitně dokumentována a vydávána pod open source licenci.

4.6 Obecný model klientské části

V této kapitole uvedu stručný přehled tříd klientské části (tj. aplikace na Raspberry Pi) a jejich účelu. Zjednodušený diagram tříd lze nalézt na obrázku 11.



Obrázek 11 - Diagram tříd klientské části aplikace (program v Raspberry Pi)

Jak bylo již zmíněno, jádrem aplikace je balík *security*. V následujícím výčtu jsou všechny třídy, které balík obsahuje a jejich stručný popis.

- Třída *Main* je vstupním bodem aplikace. Jejím účelem je provést uživatele úvodním nastavením a postarat se o zpracovávání snímaných událostí.
- Třída *GUI* vytváří grafické uživatelské rozhraní pro uživatele a umožňuje ovládání aplikace.
- Třída *SecurityEvent* vytváří objekty pro každou událost, kterou jsme vyhodnotili jako zajímavou. Tyto události se budou ukládat na disk a prostřednictvím internetu posílat na server.
- Třída *StartupSettings* slouží k uchovávání uživatelem stanoveného nastavení chování aplikace.

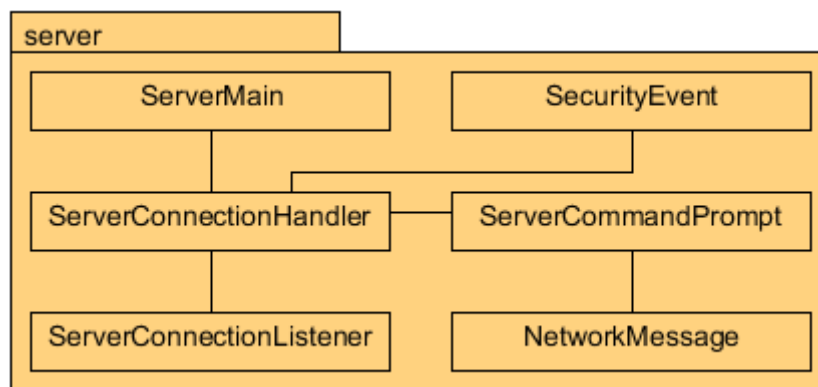
O síťovou komunikaci se stará balík *client*, který kromě přijímání a odesílání zpráv také navazuje a ukončuje spojení. Stejně jako u balíku *security* si uvedeme výčet tříd a jejich stručný popis.

- Třída *ClientConnectionHandler* navazuje a ukončuje spojení. Také poskytuje příkazy pro odesílání zpráv.
- Třída *ClientConnectionListener* přijímá zprávy ze serveru, interpretuje je a předává hlavní třídě *Main*.
- Třída *ClientCommandPrompt* umožňuje uživateli odeslat příkaz na server prostřednictvím příkazové řádky.
- Třída *NetworkMessage* slouží k vytváření zpráv, které se dají posílat mezi serverem a klientem.

Projekt využívá knihovnu *Sarxos Webcam Capture*, ze které využíváme třídy *Webcam*, *WebcamPanel* a *WebcamMotionDetector*.

4.7 Obecný model serverové části

Serverová část aplikace se stará o příjem zpráv (události a příkazy) a o ukládání snímků na příslušná místa na disku. Server bude také třídit přijaté snímky do podsložek podle jejich data a času. Výstupem serveru bude log zaznamenaných událostí pro přehlednější orientaci. Zjednodušený diagram tříd si můžeme prohlédnout na obrázku 12.



Obrázek 12 - Diagram tříd pro server

Třídy fungují na stejném principu jako v balíku *client* a bylo by nadbytečné je zde popisovat znova.

4.8 Web

Webové rozhraní je velmi jednoduché, obsahuje náhled na aktuální snímek s popisem data a času pořízení, tlačítko aktualizovat, uložit aktuální obrázek, výpis posledních událostí a tlačítko pro zobrazení složek s nasnímanými daty. Náhled webu můžeme vidět na obrázku 13.



Obrázek 13 – Ukázka webového rozhraní

5 Předpokládané a dosažené výsledky

V jednotlivých podkapitolách porovnáme předpokládané výsledky s výsledky dosaženými. Předpokládané výsledky byly hrubým odhadem nebo jen návrhem před samotnou realizací a proto zřídka kdy dosahujeme očekávaného výstupu práce. Součástí porovnání je i diskuse možných vylepšení a alternativ.

5.1 Snímkovací frekvence a rozlišení

Snímkovací frekvence a rozlišení byly již od začátku hlavním pilířem celé práce. Pokud by se nám nepodařilo dostat rozumnou snímkovací frekvenci při rozlišení, které je pro monitorování použitelné, tak by výstupem projektu bylo, že Raspberry Pi je pro monitorování nepoužitelné. Abychom zjistili dostupná rozlišení a jejich předpokládané snímkovací frekvence, existuje příkaz, který vypíše dostupná rozlišení, které ovladač médií na Raspberry Pi podporuje a k nim příslušnou maximální snímkovací frekvenci. Příkaz vypadá takto:

```
v4l2-ctl --list-formats-ext
```

Po zadání příkazu do terminálu Raspbianu získáme tento výstup:

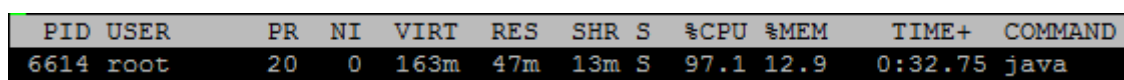
```
Index      : 1
Type       : Video Capture
Pixel Format: 'YUYV'
Name       : YUV 4:2:2 (YUYV)
Size: Discrete 640x480
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 160x120
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 176x144
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 320x240
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 352x288
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 640x360
           Interval: Discrete 0.033s (30.000 fps)
Size: Discrete 800x600
           Interval: Discrete 0.050s (20.000 fps)
Size: Discrete 960x720
           Interval: Discrete 0.067s (15.000 fps)
Size: Discrete 1024x576
           Interval: Discrete 0.125s (8.000 fps)
Size: Discrete 1280x720
           Interval: Discrete 0.125s (8.000 fps)
Size: Discrete 1280x960
           Interval: Discrete 0.200s (5.000 fps)
Size: Discrete 1392x768
           Interval: Discrete 0.125s (8.000 fps)
Size: Discrete 1600x896
           Interval: Discrete 0.200s (5.000 fps)
Size: Discrete 1920x1080
           Interval: Discrete 0.200s (5.000 fps)
```

Obrázek 14 - Maximální snímkovací frekvence při daných rozlišeních

Pro monitorování obecně stačí 1 až 5 snímků za sekundu. Tento předpoklad by podle výpisu znamenal, že můžeme použít libovolné rozlišení. Po několika jednoduchých experimentech jsem zjistil, že z webkamery *Defender G-lens 2693 FullHD*²⁶ získáme pouze 3 až 5 snímků za sekundu při rozlišení 320 na 240 bodů. Mnoho dalších uživatelů nahlašovalo tento problém na oficiálním fóru²⁷ a z uživatelských zkušeností plyne, že problém je následkem nedostatečného napájení kamery, které je zapříčiněno úsporným návrhem Raspberry Pi. Řešením by bylo využívat kamerový modul nebo opatřit kameru samostatně napájeným USB hubem. Celkově je rozlišení i snímkovací frekvence dostačující, bohužel jsme nedosáhli teoretických předpokladů.

5.2 Nároky na paměť a procesor

Raspberry Pi disponuje 512MB RAM a nepředpokládá se, že by na něm běžela ještě jiná náročná aplikace. Celkově ale aplikace při svém provozu nepřekročila 180MB alokované paměti. Na obrázku 15 můžeme vidět systémové nároky klientské části systému v běžném provozu.



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6614	root	20	0	163m	47m	13m	S	97.1	12.9	0:32.75	java

Obrázek 15 – Systémové nároky klienta (OS Raspbian)

Z obrázku 14 je také vidět, že proces vytěžuje procesor na 97,1%. Aplikace se snaží využít veškerý výkon, který je poskytnut operačním systémem. Ani takto vytížené Raspberry Pi se po šesti hodinách provozu nezahřívá více než na obvyklou provozní teplotu. Běh ostatních aplikací není kriticky narušen, ale snížení výkonu je znatelné.


Procesor Raspberry Pi jsem zkoušel přetaktovat na 900MHz, které výrobce stále udává za bezpečné. Návod na přetaktování a užitečné informace jsou dostupné v [15]. Protože přetaktování nevedlo ke znatelně lepším výsledkům a tak

²⁶ <http://www.czc.cz/defender-g-lens-2693-fullhd/148608/produkt>

²⁷ <https://www.raspberrypi.org/forums/viewtopic.php?f=44&t=27118>

přetaktování procesoru mohu doporučit pouze těm, kteří nebudou využívat bateriové napájení.

Paměťové nároky serverové části systému při plném vytížení nepřesahují 100MB. Na obrázku 16 můžeme vidět, jak server vytěžuje systém během běžného provozu. V serverové části se vyskytl problém, kdy server krátce po spuštění spadl a jako důvod uvedl vyčerpanou paměť. Tento *memory leak*²⁸ byl způsoben chováním metod *ObjectInputStream.readObject()* a *ObjectOutputStream.writeObject()*, které se starají o příjem a odesílání snímků. Tyto metody uchovávají reference na přijaté a odeslané zprávy a tak je *Garbage Collector* neuvolňuje. Řešením je využití metod *readUnshared()* a *writeUnshared()*, které neuchovávají reference na přijaté a odeslané objekty, což vede k uvolňování paměti a eliminaci problému s pádem serveru. Další informace o programování síťových aplikací v Javě lze získat v [16].

Název	Stav	Procesor	Paměť	Disk	Sít
 Java(TM) Platform SE binary		0 %	29,9 MB	0 MB/s	0 Mb/s

Obrázek 16 - Systémové nároky serveru (OS Windows 8.1)

Celkové systémové nároky obou částí jsou přijatelné a aplikace mají dostatečnou rezervu i při maximálním vytížení.

5.3 Cena a efektivita

Cenu systému spočítáme součtem jednotlivých použitých komponent. V součtu není zahrnuta pořizovací cena počítače pro server, protože se předpokládá, že takový počítač je běžnou součástí vybavení domácnosti. Veškerý použitý software je zdarma. Uvažujeme tedy tedy cenu Raspberry Pi model B+ které v době realizace stálo 880 Kč²⁹ a cenu webkamery *Defender G-lens 2693 FullHD* která v době realizace stála 542 Kč, což je v součtu 1422 Kč. Aktuálně nejpodobnější konkurenční produkt jsem našel kamerový systém *Swann SWADW-410CAM*³⁰ v ceně 101,54 USD, což je v této době přibližně 2487 Kč. Protože ceny kamerových systémů vyšších kvalit se

²⁸ Termín užívaný pro nežádoucí rostoucí nároky programu na paměť. Jinými slovy - únik paměti.

²⁹ <http://rpishop.cz/raspberry-pi-pocitace/74-raspberry-pi-7568308.html>

³⁰ <http://www.walmart.com/ip/Swann-SWADW-410CAM-US-Extra-Camera/38333928>

pohybují až v řádech desetitísiců, tak náš systém je na tom cenově výborně. Tato skutečnost vyhověla našim předpokladům a motivaci celého projektu.

Konkurenční produkty jsou oproti našemu systému velmi jednoduché na instalaci, obsluhu, konfiguraci a jek nim dostupná technická podpora. Konkurenční produkty také ve většině případů dosahují lepších výsledků v oblasti kvality snímků, nebo nabízejí širší paletu funkcí např. noční vidění, široký záběr apod. Efektivita konkurenčních produktů může spočívat v tom, že vyrábí specializovaný, jednoúčelový hardware, oproti tomu Raspberry Pi je univerzální a přizpůsobitelné. Efektivitě našeho systému také ubírá složitější instalace a zmíněné limitace snímkovací frekvence a rozlišení kamery.

6 Zhodnocení práce a získané zkušenosti

V rámci této bakalářské práce vznikl jednoduchý prototyp monitorovacího systému pro počítač Raspberry Pi. Jelikož se jedná o prototyp, chybí systému některé funkce, které mohou být námětem pro rozšíření stávající práce, viz kapitola 7. Hlavním problémem aplikace v aktuálním stavu je kvalita snímků, konkrétně snímkovací frekvence a rozlišení, které jsou dostatečné, ale nevyužívají celý potenciál systému.

Grafické uživatelské rozhraní na Raspberry Pi je nadbytečné a nepraktické. Uživatel musí připojit monitor a myš, aby mohl sledovat živý přenos. Jediné možné využití implementovaného GUI je při prvotním nastavení, ale i tak se bude hůře manipulovat s monitorem a kabely, pokud budeme umisťovat Raspberry Pi na vyvýšená místa.

Periferie připojitelné na V/V piny jako LED, piezo atd. jsem otestoval a vytvořil jednoduchý senzor pro monitorování dveří, oken, skříní atd. (viz část 4.5). Systém je prakticky použitelný a je vhodný pro monitorování zmíněných objektů (malé obchody, trafiky, dílny, garáže atd.).

Při práci s Raspberry Pi jsem se naučil efektivně pracovat s operačním systémem Raspbian a seznámil jsem se s novými operačními systémy, jako např. Plan 9 nebo Risc OS. Dále jsem získal zkušenosti v oblasti návrhu aplikací, či zabezpečovacích systémů a optimalizace programů na počítačích s omezenými zdroji. Mezi důležité poznatky také patří, že při dostatečné dekompozici problému

se dají řešení jednotlivých podproblémů dohledat na internetu (např. v tomto projektu jsem nemusel řešit jak získat v programu snímek z webkamery díky použité knihovně, což umožnilo lépe se soustředit na samotné zadání). Výsledné programové kódy jsem umístil na GitHub³¹ pro snadnější přístup.

7 Doporučení pro další vývoj

Během práce vznikl funkční prototyp monitorovacího systém, který může sloužit jako základ pro budoucí rozšíření. Rozšíření jsem rozdělil do dvou podkapitol podle priority.

7.1 Doporučení s vysokou prioritou

Hlavním nedostatkem aplikace je nízká kvalita pořizovaných snímků, proto by bylo vhodné prozkoumat kamerový modul od výrobce Raspberry Pi a zakomponovat ho do projektu. Vývojář knihovny *Sarxos Webcam Capture* přidal 31. března 2015 podporu pro kamerový modul, takže stačí drobná úprava zdrojových kódů³² naší aplikace.

Během návrhu aplikace jsem se snažil odstínit uživatele od pokročilých nastavení, abych minimalizoval počet úkonů nutných ke zprovoznění aplikace. Dalším možným rozšířením by bylo přidání „pokročilých nastavení“ pro ovládání například citlivosti detekce pohybu.

7.2 Doporučení s nízkou prioritou

Pokud by se podařilo získat větší snímkovací frekvenci z kamery, tak vhodným rozšířením by byl streamovaný přenos např. protokolem RTSP (*Real Time Streaming Protocol*). Toto rozšíření by však zvýšilo systémové nároky na Raspberry Pi a zatížilo by síťový přenos, proto ho zařazuji do kategorie s nízkou prioritou.

Dalším možným rozšířením bylo naimplementovat podporu serveru pro více Raspberry Pi. Program jsem psal tak, aby toto rozšíření bylo snadno realizovatelné, ale sám jsem jej neimplementoval z důvodu nadbytečnosti. V systému by stačilo dodělat, aby každé Raspberry Pi s odesílanými snímky posílalo i svůj jedinečný identifikátor a server by se postaral o třídění snímků podle zdrojového zařízení.

³¹ <https://github.com/Smokedealer/Raspberry-Pi-Security>

³² <https://github.com/sarxos/webcam-capture/issues/286>

Toto rozšíření by vyžadovalo i rozšířit web o prohlížení snímků z různých zařízení, případně zavedení databáze.

8 Závěr

Cílem projektu bylo prozkoumat možnosti využití počítače Raspberry Pi v oblasti monitorování a vytvořit jednoduchý prototyp, který využití demonstruje. V textu práce byly diskutovány dostupné operační systémy a programovací jazyky pro Raspberry Pi a stejně tak jeho využití pro monitorovací a ostatní účely.

Cíle stanovené na začátku práce a požadavky upřesňující výslednou podobu práce byly splněny v plném rozsahu. K monitorování jsem využil webkameru i V/V piny. Aplikace je nezávislá na druhu použité webkamery. Uživatel může vyvolat požadavek na Raspberry Pi prostřednictvím internetu a prohlížet nasnímaná data prostřednictvím webového prohlížeče. Jak bylo již diskutováno, sledování přenosu na webu není v aktuálním stavu plynulé a tak jedním z možných vylepšení by bylo použití protokolu RTSP pro přenos obrazu. Několika jednoduchými experimenty jsem ověřil, že je systém prakticky použitelný. Raspberry Pi je tedy vhodné pro monitorovací účely.

Reference

- [1] SCHMIDT, Maik. *Raspberry Pi: A Quick-Start Guide*. El Dorado: The Pragmatic Programmers, 2012. ISBN: 978-1-93778-504-8.
- [2] MCMANUS, Sean a Mike COOK. *Raspberry Pi For Dummies*. New York: John Wiley & Sons, 2013. ISBN: 978-1118554210.
- [3] RICHARDSON, Matt a Shawn WALLACE. *Make: Getting Started with Raspberry Pi*. Sebastopol: O'Reilly Media, 2014. ISBN: 9781457186127.
- [4] GIRLING, Gary. *Raspberry Pi: a practical guide to the revolutionary small computer*. Sparkford: Haynes Publishing, 2013. ISBN: 9780857332950.
- [5] PILCH, Tomasz. *Možnosti použití Raspberry Pi pro domácí automatizaci*. Brno, 2014. Bakalářská práce. Vysoké učení technické v Brně.
- [6] ROBINSON, Andrew a Mike COOK. *Raspberry Pi Projects*. John Wiley & Sons, 2013. ISBN: 978-1118555439.
- [7] WASILIK, Stan. *Essential home security: a layman's guide*. New York: CreateSpace Independent Publishing Platform, 2010. ISBN: 978-1453732038.
- [8] LYON, David. *Surveillance studies: on overview*. Malden, MA: Polity, 2007. ISBN: 978-0745635927.
- [9] MEMBREY, Peter a David HOWS. *Learn Raspberry Pi with Linux (Technology in Action)*. New York: Apress Publishers, 2012. ISBN: 9781430248217.
- [10] MONK, Simon. *Programming the Raspberry Pi: getting started with Python*. New York: McGraw-Hill, 2013. ISBN: 978-0071807838 .
- [11] HEROUT, Pavel. *Učebnice jazyka Java*. 5. rozš. vyd. České Budějovice: Kopp nakladatelství, 2011. ISBN: 9788072323982.

- [12] PECINOVSKÝ, Rudolf. *Java 8: úvod do objektové architektury pro mírně pokročilé*. 1. vyd. Praha: Grada, 2014. ISBN: 978-80-247-4638-8.
- [13] FIRYN, Bartosz. n. d. Webcam Capture API. In: *Webcam Capture* [online]. [cit. 2015]. Dostupné z: <http://webcam-capture.sarxos.pl/>
- [14] SAVAGE, Robert. n. d. *The Pi4J Project - Home* [online]. [cit. 2015]. Dostupné z: <http://pi4j.com/>
- [15] UPTON, Eben. n. d. *Introducing Turbo Mode* [online]. [cit. 2015]. Dostupné z: <https://www.raspberrypi.org/introducing-turbo-mode-up-to-50-more-performance-for-free/>
- [16] HAROLD, Elliotte Rusty. *Java network programming*. 4th edition. Beijing: O'Reilly, 2014. ISBN: 978-144-9357-672.

Příloha - Uživatelská příručka

Překlad aplikace

Překlad klientské části i serverové části je dostupný pomocí nástroje *Ant*. Pro překlad je potřeba být v adresáři, kde se nachází soubor *build.xml* a adresář *src*, který obsahuje zdrojové kódy aplikace. Pro překlad klientské části slouží příkaz:

```
ant
```

Pro překlad serverové části použijeme příkaz:

```
ant server
```

Mějme na paměti, že k překladu mohou být vyžadována administrátorská práva. Spustitelné soubory formátu *.jar* se ukládají do adresáře *jar*.

Instalace a konfigurace

Po úspěšném přeložení jsou klientská a serverová část již spustitelné. K instalaci webové části je potřeba nakopírovat soubory ze složky *web* do kořenového adresáře (či nějaké jeho podsložky) webserveru, který je na daném počítači spuštěn.

Konfigurace klienta probíhá při prvním spuštění automaticky pomocí interaktivního průvodce. Průvodce nastavením zjišťuje jen nezbytně nutné informace a dá se opětovně vyvolat spuštěním s parametrem *-r*:

```
java -jar monitor.jar -r
```

Konfigurace serveru probíhá automaticky, chce-li uživatel nastavit jiný port, na kterém bude server naslouchat, dá se tak učinit zadáním parametru *-p* následovaného hodnotou žádaného portu např.:

```
java -jar server.jar -p 10001
```

Web je implicitně přednastaven, aby fungoval po nakopírování do kořenové složky (nebo podsložky) webserveru. Budou-li se snímky přijaté serverem ukládat na jiné úložiště než do předpřipravených složek webu, je zapotřebí upravit zdrojové kódy webu a proto tento postup nedoporučuji.

Spuštění

Chceme-li odesílat snímky na server, je potřeba spustit server jako první, než se k němu připojí klient. Server spustíme příkazem:

```
java -jar server.jar
```

Implicitně je server nastavený, aby naslouchal na portu 10000. Pro změnu portu viz instalace a konfigurace. Po spuštění serveru lze nyní připojit klienta. Pro spuštění klienta zadáme příkaz:

```
java -jar monitor.jar
```

Ovládání

GUI klienta umožňuje náhled na živý přenos z kamery v levé části okna. V pravé části okna máme tlačítko pro uložení aktuálního snímku, stavové výpisy jako název kamery, aktuální snímkovací frekvence, rozlišení, připojený server, port serveru a seznam posledních 20 zaznamenaných událostí. Klikneme-li na položku v seznamu událostí, okamžitě se náhled živého záznamu zastaví a na jeho místě se zobrazí snímek zvolené události. Kliknutím mimo pole událostí znovu aktivujeme zobrazení živého přenosu. Příkazy serveru jsou dostupné zadáním příkazu *help* do příkazové řádky, kde je server spuštěný. Tyto příkazy slouží primárně k ovládání Raspberry Pi.

Webová stránka kvůli zmíněným limitacím nezobrazuje živý přenos v reálném čase, ale je periodicky aktualizována. Pro uspíšení aktualizace slouží tlačítko *Refresh*. Uložení aktuálního obrázku je možné stiskem tlačítka *Save Image*. Zaznamenané události lze prohlížet po stisknutí tlačítka *Archive*. Události jsou seříděné podle dne pořízení do podsložek a následně podle času pořízení do jednotlivých souborů.