

Primjena Raspberry Pi u obrazovanju

Novosel, Dino

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:407234>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported / Imenovanje-Nekomercijalno-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-11-26**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Dino Novosel

**PRIMJENA RASPBERRY PI U
OBRAZOVANJU**

DIPLOMSKI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Dino Novosel

Matični broj: 0119013386

Studij: Informacijsko i programsko inženjerstvo

PRIMJENA RASPBERRY PI U OBRAZOVANJU

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Dragutin Kermek

Varaždin, rujan 2020.

Dino Novosel

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj diplomski rad istraživanje je mogućnosti sklopovskih komponenti i programskih paketa Raspberry Pi računala. U radu obrađene su osnove Linux operacijskog sustava kao podloga korištenju ostalih operacijskih sustava baziranih na Linux jezgri. Osim operacijskih sustava baziranih na Linux-u u radu proći će se i kroz specijalizirane operacijske sustave razvijene isključivo za ARM arhitekture procesora. Zajednička značajka svih sustava je maksimalno iskorištavanje potencijala Raspberry Pi računala. Osim operacijskih sustava naglasak je stavljen i na dostupne programske jezike, kao i osnovne programski pakete i integrirana razvojna okruženja. Obradeno štivo ne ide u dublje analize rada pojedinih sustava, već je edukativnog karaktera i omogućuje početnicima savladavanje gradiva. Na kraju rada nalazi se programsko rješenje pojednostavljenog modela pametnog sustava koji svoju primjenu može pronaći u obrazovnim ustanovama. Programsko rješenje implementirano je korištenjem raznih senzora čija funkcionalnost i način integracije predstavljaju jednu od mogućnosti korištenja računala u podizanju kompetencija studenata i općenito primjene Raspberry Pi računala u obrazovanju. Na temelju instaliranog rješenja izneseni su osobni stavovi i zaključci o prednostima i manama Raspberry Pi računala. Rad je napravljen u okviru Laboratorija za web arhitekture, tehnologije, servise i sučelja.

Ključne riječi: Raspberry Pi, sklopovlje, operacijski sustavi, programski alati, programski jezik

Sadržaj

Sadržaj.....	iii
1. Uvod.....	1
2. Metode i tehnike rada	2
2.1 Struktura rada	2
3. Raspberry Pi.....	3
3.1 Sklopovlje	4
3.1.1. Napajanje.....	6
3.1.2. Radna memorija – LPDDR SDRAM	7
3.1.3. Procesor.....	7
3.1.4. Mrežne komponente.....	8
3.1.5. Ulazi / izlazi opće namjene – GPIO	9
3.1.6. Ostali sklopovski moduli	9
3.2. Linux platforma	11
3.2.1. Prednosti i mane Linux operacijskog sustava.....	11
3.2.2. Arhitektura operacijskog sustava.....	13
3.2.3. Struktura direktorija	14
3.2.4. Osnovne naredbe.....	16
3.3. Udaljeni pristup	21
3.3.1. SSH protokol.....	21
3.3.2. SFTP protokol	22
3.3.3. SCP protokol.....	23
3.3.4. VNC protokol.....	24
4. Operacijski sustavi.....	25
4.1. Raspbian.....	27
4.1.1. Instalacija operacijskog sustava	27
4.1.2. Konfiguracija sustava	28

4.1.3. Korisnički programi.....	29
4.1.4. Značajke operacijskog sustava	29
4.2. Ubuntu MATE	29
4.2.1. Instalacija operacijskog sustava	30
4.2.2. Konfiguracija sustava	30
4.2.3. Korisnički programi.....	31
4.2.4. Značajke operacijskog sustava	31
4.3. Ubuntu Server.....	31
4.3.1. Instalacija operacijskog sustava	32
4.3.2. Konfiguracija sustava	32
4.3.3. Korisnički programi.....	32
4.3.4. Značajke operacijskog sustava	32
4.4. RiscOS.....	33
4.4.1. Instalacija operacijskog sustava	34
4.4.2. Konfiguracija sustava	34
4.4.3. Korisnički programi.....	34
4.4.4. Značajke sustava	34
4.5. BerryBoot.....	35
4.5.1. Instalacija operacijskog sustava	35
4.5.2. Konfiguracija sustava	36
4.5.3. Korisnički programi.....	36
4.5.4. Značajke sustava	36
5. Programski jezici i alati za razvoj programskih rješenja na Raspberry Pi	37
5.1. Podržani programski jezici	38
5.1.1. Programski jezik C	38
5.1.2. Programski jezik C++	40
5.1.3. Python.....	42
5.1.4. Java	44
5.2. BlueJ.....	46

5.3. GeanyIDE	47
5.4. Scratch.....	48
6. Programsko rješenje	50
6.1. Slučajevi korištenja	50
6.1.1. Slučaj korištenja za RFID modul.....	51
6.1.2. Slučaj korištenja za senzor plina	51
6.1.3. Slučaj korištenja za senzor vlage i temperature.....	51
6.1.4. Slučaj korištenja za senzor plamena	52
6.1.5. Slučaj korištenja za senzor kiše	52
6.1.6. Slučaj korištenja za fotoosjetljivi senzor, senzor pokreta i relej.....	53
6.2. Senzori.....	53
6.2.1. DHT11 – digitalni senzor temperature i vlage.....	53
6.2.2. MQ-2 – senzor plina	54
6.2.3. YL-83 - Senzor za kišu	55
6.2.4. KY-026 – Senzor plamena	56
6.2.5. MF-RC522 – RFID modul.....	56
6.2.6. GL55 – fotoosjetljivi senzor	57
6.2.7. HC-SR501 – senzor pokreta	58
6.2.8. Dvokanalna elektromagnetna sklopka.....	58
6.3. Instalacija i konfiguracija alata za razvoj programskog rješenja.....	59
6.3.1. Docker.....	60
6.3.2. MySQL Workbench	62
6.3.3. NetBeans	64
6.4. Arhitektura sustava	65
6.4.1. Redis.....	66
6.4.2. MySQL.....	68
6.4.3. Programsko rješenje – Python	69
6.4.4. Programsko rješenje – Java	73
6.5. Grafičko sučelje	76

6.6. Dodatne mogućnosti - Grafana	80
7. Prednosti i nedostaci Raspberry Pi računala.....	82
7.1. Prednosti Raspberry Pi računala.....	82
7.2. Nedostaci Raspberry Pi računala	83
8. Zaključak	84
Popis literature	85
Popis slika	87
Popis tablica.....	90

1. Uvod

Raspberry Pi svojom pojavom na tržištu donio je potpunu revoluciju u načinu implementacije i razvoju „uradi sam“ projekata. To su obično projekti koji ne iziskuju ogromnu procesorsku niti memorijsku snagu, već iskorištavaju sve blagodati koje može pružiti jedno računalo malog formata. Entuzijazam korisnika Raspberry Pi računala potakla je u prvom redu prenosivost, zatim jednostavnost i podrška. Višestruke mogućnosti povezivanja sa ostalim računalima i senzorima na mreži podigle su kreativnost korisnika i pripomogle stvaranju zajednice interneta stvari. Ovaj gotovo nevjerojatan uspjeh i popularnost u početku stvaranja kreatori Raspberry Pi računala mogli su samo sanjati. Prvotna ideja projekta bila je stvaranje jednostavnog računala koje bi pomoglo u podizanju računalne pismenosti zemljama u razvoju.

Stoga su uvjeti da računalo mora biti pristupačno, mora podržavati postojeću periferiju i na kraju svega mora omogućavati razvoj računalnih programa u nekoliko programskih jezika bili osnovna nit vodilja pri stvaranju ovakvog proizvoda. Prednosti koje nudi jedan ovakav integrirani sustav brzo su uočili računalni entuzijasti iz cijeloga svijeta i ubrzo je Raspberry Pi postalo najprodavanije Britansko računalo. Vođeni uspjehom i konstantnim napretkom tehnologije do trenutka pisanja rada kreatori Raspberry Pi računala uspješno su na tržište plasirali i četvrtu inkarnaciju ovog stvarno nevjerojatnog malog komada sklopovlja.

2. Metode i tehnike rada

Diplomski rad baziran je na upoznavanju i korištenju cjelokupnog ekosustava Raspberry Pi računala, te njegovo korištenje u svrhu obrazovanja. Za pisanje rada korišteni su primarni izvori podataka u obliku znanstvene literature i akademskih članaka, dok su sekundarni izvori podataka temeljeni na internetskim stranicama, osobnim stavovima korisnika Raspberry Pi računala, kao i osobnom stavu autora rada stečenom prilikom korištenja računala u svrhu izrade aplikacijskog rješenja. Literatura je navedena redoslijedom pojavljivanja u tekstu.

Aplikacijski paketi korišteni za izradu pisanog dijela rada su Linux operacijski sustav u distribuciji Kali Linux, operacijski sustavi za Raspberry Pi: Raspbian, Ubuntu Mate, Ubuntu Server i RiscOS te integrirana razvojna sučelja: BlueJ, GeanyIDE i Scratch. Od programskih jezika za demonstraciju osnovnih funkcionalnosti korišteni su C, C++, Python i Java.

2.1 Struktura rada

Diplomski rad sastoji se od 8 poglavlja koja objašnjavaju glavne koncepte, dok svako od pod poglavlja detaljnije ulaze u srž i dodatno objašnjavaju važnije elemente teme poglavlja. Prvo poglavlje je uvod koji opisuje glavnu ideju Raspberry Pi računala. Drugo poglavlje odnosi se na metode i tehnike rada korištene pri izradi rada, objašnjava izvore literature i korištene aplikacijske pakete. Treće poglavlje vezano je za samo Raspberry Pi računalo, sklopovlje i Linux operacijski sustav. Četvrto poglavlje daje uvid u najpoznatije operacijske sustave korištene na Raspberry Pi računalu i njihove glavne značajke. Peto poglavlje bavi se programskim jezicima i alatima za efikasnu manipulaciju programskim kodom. Šesto poglavlje programsko je rješenje izvedeno na Raspberry Pi računalu, a uključuje korištenje senzora za implementaciju pojednostavljenog modela pametnog sustava aplikabilnog u mnogim obrazovnim institucijama, koje je u mogućnosti ovisno o ulaznim parametrima aktivno reagirati na promjene i na adekvatan način tražiti interakciju od korisnika. Sedmo poglavlje objašnjava prednosti i mane Raspberry Pi računala. Zadnje poglavlje je zaključak donesen na temelju cjelokupnog doživljaja stečenog tijekom korištenja Raspberry Pi računala i pisanja rada.

3. Raspberry Pi

Raspberry Pi je pristupačno, jeftino računalo izrazito malih dimenzija koje se funkcionalnostima nikako ne razlikuje od standardnog stolnog računala. Za prosječnog računalnog korisnika od standardnog stolnog računala distancira se ponajprije dizajnom, naime sve komponente Raspberry Pi računala smještene su na jednu tiskanu pločicu za razliku od stolnog računala gdje su komponente kao što su procesor, grafička kartica ili pak zvučna kartica izvedene individualno, a računalo kao cjelinu čine kada se sve komponente povežu preko matične ploče.

Treba naglasiti da se Raspberry Pi računalo prodaje bez dodatne opreme kao što je kućište ili pak napajanje. Korisniku se ostavlja odabir što sve želi povezati sa računalom pa je sukladno tome primoran nabaviti odnosno kupiti dodatnu opremu. Na početku bitno je znati da osim što je moguće birati dodatnu opremu, moguće je odabrati i sam model računala. Naime od kada se računalo pojavilo na tržištu moguće je odabrati „A“ odnosno „B“ verziju računala. Verzije računala razlikuju se po performansama što je izravan rezultat ugrađenih komponenti koje se nalaze na samoj tiskanoj pločici. Za sada, dovoljno je ukratko reći da su verzije sa oznakom „A“ općenito slabije opremljene od verzija „B“, dok će u kasnijim poglavljima biti detaljnije objašnjene razlike. Nadalje prirodno se nameće pitanje koji model onda odabrati? To najviše ovisi o zahtjevima korisnika, ali kako je riječ o računalu koje se nalazi u cjenovnom rangu jednog ozbiljnijeg obroka, a ne osobnog računala slobodno se može reći da će samo pravi entuzijasti i poznavaoči tehnologije imati dovoljno dobre argumente za odabrati jednu od ove dvije verzije dok će prosječan korisnik bez previše razmišljanja odabrati jači model.

Bez obzira koji model odabrao, korisnik ima mogućnost spajanja periferije kao što su tipkovnica ili miš, odnosno zaslon televizora ili računalni monitor. Bez ovih osnovnih mogućnosti samo računalo ne bi imalo smisla, jer korisnik ne bi mogao ostvarivati dvosmjernu komunikaciju sa računalom, bilo da se radi o običnom pretraživanju interneta ili naprednijim stvarima poput računalnog programiranja. Na tržištu naime postoje i „računala“ odnosno mikro upravljači koji korisniku omogućuju izvođenje određenih računalnih programa bez prikaza na ekranu, ali oni neće biti u fokusu ovoga rada pa se čitatelju savjetuje da detaljnije informacije o takvim računalima potraži na internetu. Kako bi se još detaljnije upoznali sa samim računalom u idućim poglavljima pobliže će se proći kroz sve sklopovske elemente Raspberry Pi 3 Model B računala, njihovim zadaćama i funkcionalnostima koje ostvaruju, te podržanim operacijskim sustavima. Raspberry Pi 3 Model B biti će korišten i u realizaciji programskog rješenja zbog njegove dostupnosti u vremenu pisanja ovoga rada.

3.1 Sklopovlje

Kako bi uopće znali kakvo je sklopovlje, odnosno o kakvom se računalu zapravo radi, na početku poglavlja slijedi upoznavanje sa osnovnim specifikacijama Raspberry Pi 3 Model B računala¹:

Tablica 1: Specifikacije Raspberry Pi 3 Model B računala

Model	Raspberry Pi 3 Model B
Arhitektura računala	ARM v8
Sustav u čipu (SoC)	Broadcom BCM2837
Procesor	1.4 GHz ARM Cortex-A53
Grafički procesor	Broadcom VideoCore IV
Radna memorija	1GB
Memorija	Podrška za Micro SDHC memorijsku karticu
Video izlazi	HDMI, Kompozitni RCA
Mrežne komponente	10/100Mbps mrežna kartica, Bluetooth 4.2 BLE , bežični mrežni adapter
Napajanje	5V putem Micro USB konektora

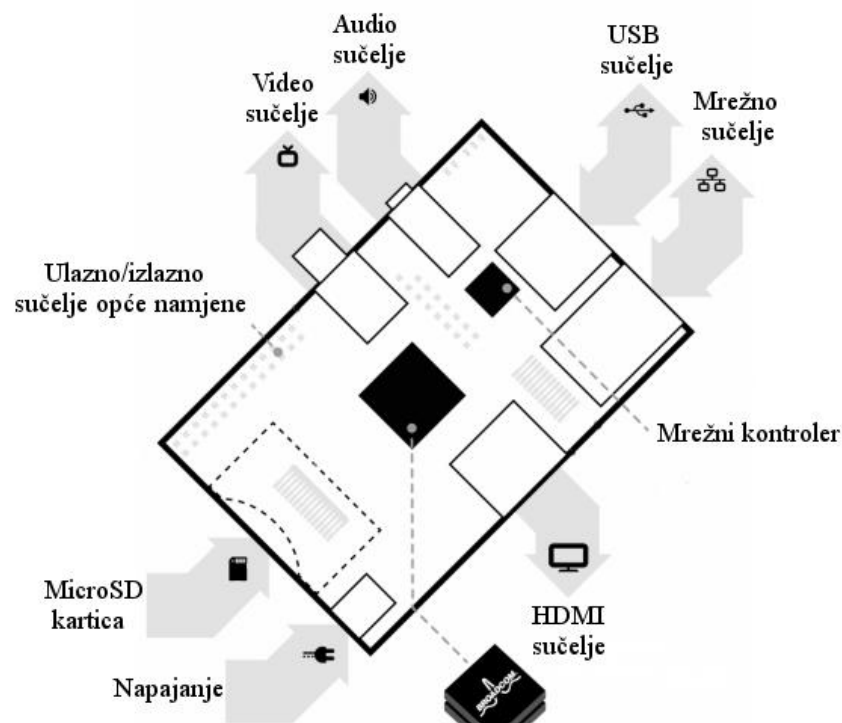
Sada kada su poznate osnovne specifikacije modela sa oznakom „B“ mogu se na primjeru jednostavno objasniti razlike u odnosu na model sa oznakom „A“ kako bi dobili dojam o kakvim se to jeftinijim rješenjima radi, odnosno koje su to komponente slabijih performansi ugrađene. Počevši od najbitnije stavke, mozga svakog računala procesora može se reći da se na ranijim modelima vidjela bitnija razlika jer su modeli sa oznakom „B“ imali procesore sa četiri fizičke jezgre dok se na jeftinijim modelima nalazilo i procesora sa jednom jezgrom, dok su se u kasnijim modelima konkretno modelima verzije 3 generacije u oba modela ugrađivali isti procesori.

Nadalje naoko vidljive razlike kod jeftinijeg modela mogu se pronaći u izostanku fizičkog Ethernet sučelja i smanjenom broju USB konektora čime se korisnika ograničilo na

¹ Raspberry Pi Foundation , Raspberry Pi 3 Model B, 2020. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (dostupno 21.04.2020.)

korištenje manje perifernih uređaja u isto vrijeme. Bitno je napomenuti da se nedostatak Ethernet sučelja može vrlo lako premostiti ugradnjom vanjske USB mrežne kartice. Nedostatak USB sučelja moguće je vrlo lako nadoknaditi povezivanjem USB središta (*eng. hub*) koje multipliciraju broj USB sučelja ovisno o željama korisnika. U ovom slučaju bitno je pripaziti na fizičke mogućnosti same implementacije elemenata na tiskanoj pločici kao što su naponski osigurači i graničnici, koji upravljaju isporukom snage prema priključenim perifernim uređajima. Pogledom dublje „ispod haube“ osim procesorske snage, razlike su vidljive u količini radne memorije koja je na jeftinijim modelima obično prepolovljena u usporedbi na „B“ modele, čime se izravno utječe na količinu i brzinu izvođenja memorijski zahtjevnijih procesa.

Prije ulaska u detaljniju analizu ugrađenih komponenti bitno je naglasiti da su sve važnije komponente koje čine računalu kao što su procesor, radna memorija, grafička kartica i ostali moduli na Raspberry Pi računalu izvedeni u konfiguraciji koja se naziva SoC (*eng. System on a Chip*) odnosno sustav na čipu. Ovakav oblik grupiranja komponenti na jedan čip izravan je rezultat napretka tehnologije i zahtjeva tržišta za malim i prenosivim uređajima, ponajprije kod pametnih telefona. Za krajnji cilj SoC čipovi imaju povećanje performansi sustava uz smanjivanje potrošene energije i lakšu integraciju na fizički medij, najčešće tiskane pločice mobilnih uređaja.



Slika 1: Shematski prikaz važnijih komponenti Raspberry Pi računala (Prema: Liz Upton, 2011, dostupno na: <https://www.raspberrypi.org/blog/new-graphic/>)

Raspored dijelova računala moguće je vidjeti na Slika 1 koja vjerno prikazuje raspored važnijeg dijela komponenti i sučelja na Raspberry Pi 3 Model B računalu. U nastavku poglavlja detaljnije će se proći kroz karakteristike i funkcionalnosti važnijih sklopova, kako bi se novim korisnicima omogućila što jednostavnija i brža integracija u ovaj računalni sustav.

3.1.1. Napajanje

Kod Raspberry Pi računala neće se pronaći klasično napajanje kao na stolnim računalima. Naći će se samo regulacijske komponente na tiskanoj pločici koje su zadužene za osiguravanje stabilnosti i postavljanja limita napona. Za napajanje računala koriste se adapteri sa micro-USB odnosno USB-C sučeljem koji se mogu pronaći i kod svih modernijih mobilnih uređaja. Ovisno o modelu Raspberry Pi računala potrebno je odabrati adapter odgovarajuće snage i odgovarajućeg sučelja. Iznosi napona koji se koriste na Raspberry Pi računalu su 5V i 3.3V. Napon od 5V zadužen je za napajanje internih komponenti računala i periferije računala, dok je niži napon od 3.3V zadužen za napajanje senzora koje je moguće putem ulazno/izlaznog sučelja opće namjene (eng. GPIO – general purpose input/output) spojiti na računalo.

Za napajanje Raspberry Pi 3 Model B računala potrebno je odabrati adapter koji je u mogućnosti isporučiti minimalno 3.5W snage. U praksi preporučuje se odabrati napajanje koje može isporučiti barem 50% više od nazivnog utroška energije trošila zbog nesavršenosti komponenata ugrađenih u samo napajanje. U ovom slučaju to znači da je za sigurnu upotrebu potrebno odabrati napajanje koje je u stanju isporučiti barem 5.25W snage. Kako ne bi došlo do zabune, deklarirana potrošnja odnosi se na samo Raspberry Pi računalo bez priključene periferije. Korištenje neadekvatnog napajanja na uređaju manifestirati će se problemima kao što su zamrzavanje slike zaslona, prestanka rada mrežnog sučelja, resetiranja uređaja prilikom paljenja i sl. Stoga se preporučuje pri korištenju novijih Raspberry Pi računala 3 i 4 generacije koristiti napajanja snage od najmanje 10W kako bi se osigurala nesmetana opskrba periferije i nesmetan rad računala.²

² S. Monk, *Raspberry Pi cookbook*. Beijing: O'Reilly, 2014.

3.1.2. Radna memorija – LPDDR SDRAM

Sinkronizirana dinamička memorija slučajnog pristupa odnosno SDRAM (*eng. synchronous dynamic random access memory*) u izvornom obliku u današnjim računalima gotovo je izumrla. Pojavila se sredinom 1990-tih godina i omogućila je sinkronizaciju radnog takta memorije sa taktom procesora. Ovom značajkom memorijski upravljač mogao je znati točno u kojem radnom taktu može zatražiti podatke, odnosno mogao je znati kada će podaci biti pripremljeni tako da ih procesor može odmah početi obrađivati bez potrebom za trošenjem dodatnih ciklusa na čekanje. U Raspberry Pi računalo ugrađena je novija generacija ove vrste memorije s oznakom LPDDR (*eng. Low Power Double Data Rate*) smanjenih energetskehtkih zahtjeva koja je prilagođena za korištenje u prijenosnim računalima i svim manjim prijenosnim uređajima, a podržava spajanje sabirnicama širine od 16, 32 i 64 bita.

U slučaju kada se govori o računalu ograničenih programskih i sklopovskih mogućnosti kao što je Raspberry Pi računalo, vrsta memorije možda i nije toliko važna koliko i sama količina. To je sigurno važan podatak koji zanima svakog razvojnog inženjera, odnosno sistemskog arhitekta. Raspberry Pi 3 Model B raspolaže sa 1GB radne memorije. Za usporedbu jeftiniji Raspberry Pi 3 Model A raspolaže sa duplo manje odnosno 512MB radne memorije. Za efikasno korištenje ograničene količine radne memorije programerima je omogućena izravna komunikacija sa ulazno izlaznim sučeljem opće namjene putem njegovih memorijskih adresa. Ovom značajkom izbjegnuto je korištenje upravljačkih programa koji mogu nepotrebno zauzimati dio radne memorije.³

3.1.3. Procesor

Raspberry Pi 3 Model B opremljen je četverojezgrenim procesorom oznake Cortex-A53 temeljnim na ARM arhitekturi. Ima varijabilni radni takt u rasponu od 600MHz do maksimalnih 1.4GHz. Model Cortex-A53 pripada novoj generaciji ARM procesora koji su po prvi puta predstavljeni u 64-bitnoj arhitekturi. Takva arhitektura omogućuje obradu veće količine podataka kao i alokaciju više radne memorije, a što je najbitnije sve to ostvaruje uz smanjenu potrošnju energije. Od karakteristika zanimljivih programerima ističe se sklopovskom

³ W. Gay, *Mastering the Raspberry Pi*. 2014.

podrškom za virtualizaciju, ugrađenim setom instrukcija za ubrzavanje kriptiranja i dekriptiranja za AES i SHA algoritme, te algoritme temeljne na eliptičnim krivuljama.⁴

Još jedna zanimljivost vezana uz procesore na Raspberry Pi uređajima je mogućnost tzv. „overclockinga“ odnosno programskog modificiranja radnog takta i napona na procesoru koji dovodi do općeg podizanja performansi rada procesora i sustava u cjelini. Ovaj se postupak u pravilu ne preporučuje neiskusnim korisnicima jer može dovesti do trajnog oštećenja procesora uslijed pregrijavanja ili općenito smanjene pouzdanosti rada sustava.

3.1.4. Mrežne komponente

Mora se priznati da u današnje vrijeme kada je sve više manje „na mreži“ i u doba Interneta stvari (*eng. Internet of things*) u kojemu sve komponente komuniciraju ili međusobno ili sa nekom vrstom servisa, uređaj bez mogućnosti spajanja na Internet bio bi skoro pa beskoristan. Raspberry Pi 3 Model B računalo opremljeno je cijelom serijom mrežnih komponenti koji omogućuju razne načine povezivanja. Na prvom mjestu nalazi se klasična mrežna kartica sa Ethernet sučeljem koja dopušta brzine prijenosa podataka do 1 Gbps što danas predstavlja nekakav standard u mrežnoj tehnologiji. Osim klasičnog mrežnog adaptera na računalu naći će se i bežični mrežni adapter koji podržava 802.11 b/g/n/ac standarde. Za one koji ne znaju 802.11 predstavlja skup standarda za bežičnu komunikaciju utemeljenih 1997. godine od strane instituta inženjera elektrotehnike i elektronike (*eng. IEEE – Institute of Electrical and Electronics Engineers*).

Najaktualniji standard u bežičnom povezivanju podržan na Raspberry Pi 3 Model B je 802.11 ac koji omogućuje simultano povezivanje na frekvencijama od 2.4GHz i 5GHz, te brzine od maksimalno 450Mbps na frekvenciji od 2.4GHz odnosno 1300Mbps na frekvenciji od 5GHz, barem u teoriji. Uz klasičnu mrežnu karticu i bežičnu mrežnu karticu računalo je opremljeno i Bluetooth modulom verzije 4.2, koji se od starije generacije ističe bržim prijenosom podataka, omogućuje veću propusnost, pojačanu sigurnost i smanjenu potrošnju energije. Po želji korisnika ostale mrežne komponente u obliku perifernih uređaja moguće je na računalo dodati putem USB sučelja.⁵

⁴ “ARM Cortex-A53 MPCore Processor Technical Reference Manual”, 2016.

⁵ A. Pajankar, “Instant Raspberry Pi,” 2016.

3.1.5. Ulazi / izlazi opće namjene – GPIO

Ulazno - izlazno sučelje opće namjene može se pronaći na svim verzijama Raspberry Pi računala i može se reći da je jedna od značajki koje ovo računalo čini prepoznatljivim. Sučelje se sastoji od 40 pinova od kojih je 26 u potpunosti korisnički upravljivo dok je ostatak predviđen za napajanje perifernih uređaja naponima od 3.3V i 5V. Svaki od 26 pinova koji su programski upravljani mogu predstavljati ulaz odnosno izlaz. Računalo prijenos podataka sa GPIO sučelja prepoznaje promjenama stanja na ulazu odnosno izlazu gdje napon od 3.3V predstavlja stanje visokog napona odnosno u binarnom sustavu predstavlja 1, dok napon od 0V predstavlja stanje bez napona i u binarnom sustavu predstavlja 0.

Stoga je ovaj način komunikacije sa priključenom periferijom prirodan, ako je poznato da sva moderna računala, osim kvantnih računala u konačnici sve programske instrukcije pretvaraju u strojni jezik odnosno u binarni oblik koji podržava dva stanja, 0 i 1. Osim jednostavnih perifernih uređaja koji mogu funkcionirati samo promjenama osnovnih stanja kao što su elektromagnetne sklopke koje načelno mogu biti u isto vrijeme ili otvorene ili zatvorene, odnosno njihovo stanje odgovara logičkoj jedinici odnosno logičkoj nuli, postoje i uređaji koji zahtijevaju kompliciranije načine komunikacije. Tako je na određenim pinovima omogućena komunikacija putem PWM (*eng. Pulse Width Modulation*) odnosno metodom modulacije širine impulsa, SPI (*eng. Serial peripheral interface*) odnosno komunikacije putem serijskog sučelja za periferiju koje podržava sinkroniziranu serijsku komunikaciju, te putem I2C (*eng. Inter-Integrated Circuit*) protokola koji predstavlja sabirnicu, koja omogućuje serijski prijenos podataka upravljani taktom.⁶

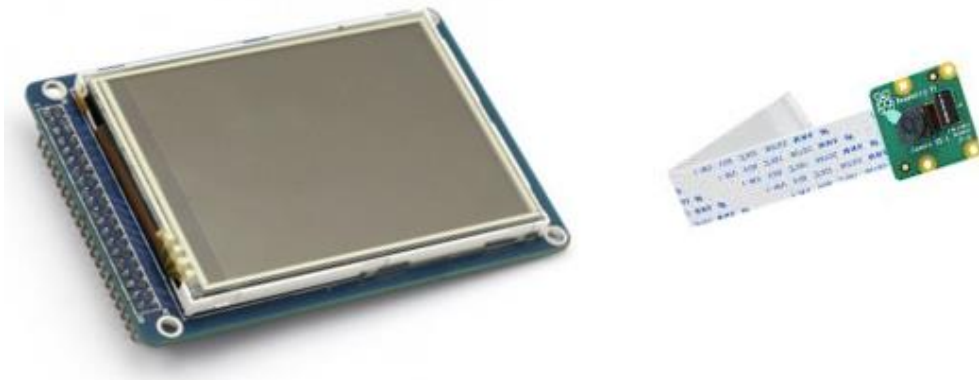
3.1.6. Ostali sklopovski moduli

Nakon detaljnijeg upoznavanja sa bitnijim sklopovskim komponentama u ovom poglavlju proći će se kroz ostatak modula koji se nalaze na računalu. Kako bi komunikacija sa računalom bila moguća na njega moramo priključiti određene periferne uređaje kojima će se zadavati instrukcije računalu. Najčešće se radi o klasičnim tipkovnicama, odnosno računalnim miševima koji zahtijevaju spajanje na računalu putem univerzalnih serijskih sučelja inače poznatijih pod kraticom USB (*eng. Universal serial bus*). Na svim verzijama Raspberry Pi računala nalazi se bar po jedno takvo sučelje, dok se na skupljim modelima može naći do čak 4 USB sučelja. Osim serijskih sučelja za komunikaciju sa računalom na svim Raspberry Pi

⁶ S. Monk, *Raspberry Pi cookbook*. Beijing: O'Reilly, 2014.

uređajima također će se naći i neki oblik sučelja za prijenos visokokvalitetnih audio video signala koji omogućuju prikaz multimedije na zaslonu, odnosno reprodukciju zvuka. Ova sučelja poznatija su pod akronimom HDMI (*eng. High definition multimedia interface*) i na Raspberry Pi računalima dolaze u 3 izvedbe: micro HDMI, mini HDMI i klasični HDMI.⁷

Sekundarna memorija računala nije izvedena u obliku tvrdog diska već je u formatu microSD kartice, što je ekvivalent modernim SSD (*eng. Solid State Drive*) diskovima u stolnim računalima. Ovaj oblik memorije osim što je dimenzijama manji od klasičnog tvrdog diska ima puno kraće vrijeme pristupa podacima što rezultira bržim radom računala i izuzetno je energetski učinkovit, što je kao što se do sada mogli zaključiti jedan od glavnih motiva svih ugrađenih komponenti na Raspberry Pi računalu. Službeno maksimalna veličina memorije koju podržava Raspberry Pi računalo je 32GB. Od ostalih sklopovskih sučelja može se spomenuti još i sučelje za zaslone osjetljive na dodir razvijene posebno za Raspberry Pi računalo, odnosno sučelje kamere koja je također posebno razvijeno za Raspberry Pi računalo. Kamera je primjenu pronašla u mnogim „uradi sam“ projektima.



Slika 2: Specijalizirana Raspberry Pi periferija - ekran osjetljiv na dodir i kamera
(Prema: <http://thepihut.com>)

⁷ G. Halfacree, *Raspberry Pi Beginner's Guide*, 2019.

3.2. Linux platforma

U ovom poglavlju naglasak će se dati na Linux operativni sustav čije izvedenice u najvećem postotku čine operacijske sustave koji se koriste na Raspberry Pi računalima. Osim operacijskih sustava baziranih na Linux-u u upotrebi mogu se pronaći i operacijski sustavi bazirani na Windows platformi ili čak operacijski sustavi specijalno kreirani za računala sa RISC arhitekturom. Linux kao operacijski sustav službeno postoji od 1991. godine kada ga je kao dodatak jezgri UNIX operacijskog sustava osmislio Linuz Torvalds. Samu UNIX jezgru osmislio je dvojac Dennis Ritchie i Ken Thompson koji stoji i iza nastanka C programskog jezika. Operacijski sustav je od samog početka zamišljen kao program otvorenog koda što znači da osim što je u potpunosti besplatan, moguće ga je dodatno nadograđivati ovisno o željama korisnika. Na osnovu toga mnogi korisnici, ali i velike kompanije razvile su svoje inačice pa tako se danas u upotrebi pronalazi više od pet stotina različitih distribucija operacijskih sustava koje svoje korijene vuku od izvorne inačice Linux-a.⁸

3.2.1. Prednosti i mane Linux operacijskog sustava

Definitivno najveća prednost Linux operacijskog sustava je njegova cijena, odnosno izostanak iste, ako se zanemare specijalizirane komercijalne verzije. Specijalizirane verzije najčešće koriste velika poduzeća za svoje poslužitelje i kao takva garantiraju konstantnu korisničku podršku uz najviši nivo sigurnosti korištenjem zadnjih sigurnosnih zakrpa. Fokus će se staviti na izdanja koja će se ipak u najvećem obimu naći na računalima običnih korisnika, a to su ona besplatna. O njima se može reći da su bazirana na GNU javnoj licenci koja kaže da svatko tko želi može slobodno preuzeti, koristiti ili nadograditi postojeće programsko rješenje sve dok programski kod i dalje bude javno dostupan. Sljedeća prednost je prenosivost odnosno kompatibilnost sa sklopovljem. Mnogi proizvođači sklopovlja iskoristili su činjenicu da je jezgra operacijskog sustava otvorenog koda i da na jednostavan i efikasan način mogu svoje proizvode učiniti potpuno funkcionalnim bez ostvarivanja dodatnih troškova. Na ovaj način upotreba Linux operacijskog sustava sa osobnih računala proširila se i na mobilne uređaje, kućanske aparate, pa čak i većinu sustava u automobilskoj industriji.

⁸ P. Cobbaut, *Linux Fundamentals*. 2015.

Kakav bi to operacijski sustav bio da ga ne krasi i značajka sigurnosti. Tek će se u iznimno rijetkim slučajevima čuti za štetu nastalu na Linux platformi kao rezultata djelovanja zlonamjernog koda. Sigurnost se na Linux platformi postiže ograničavanjem prava pristupa korisniku do funkcionalno bitnih dijelova operacijskog sustava kao što su upravljački programi, odnosno određeni dijelovi jezgre koje bi eventualni napadači mogli iskoristiti za zlonamjerne radnje na računalu. Sigurnosti i pouzdanosti svjedoči i postotak koji govori da preko 90% svih poslužitelja, kao i preko 85% svih pametnih telefona kao osnovu koristi neki oblik Linux operacijskog sustava. Kako bi bilo moguće koristiti ga na raznim platformama mnogo truda uložilo se u skalabilnost ove platforme, naime korisnik je u mogućnosti osim jezgre operacijskog sustava, a ovisno o dostupnom sklopovlju i memoriji instalirati samo one pakete koji su mu neophodni za rad. Dovoljno je reći da neke distribucije Linux-a stanu i na već sada davno zaboravljene diskete, odnosno reda veličine su otprilike 1MB. Od ostalih značajki koje dijeli sa ostalim operacijskim sustavima na popisu nalaze se: mogućnost simultanog izvođenja više procesa podjelom procesorskog vremena, mogućnost instalacije raznih korisničkih programa iz službenih biblioteka, podrška za tzv. živo pokretanje sustava sa memorijskih kartica odnosno CD-a bez instalacije i sl.⁹

Svaki proizvod ima mane pa tako i ovaj. Zbog svoje dostupnosti i činjenice da je sustav besplatan, mnogi korisnici su napravili vlastite verzije operacijskih sustava temeljenih na Linux-u, pa tako da ponovim danas postoji preko pet stotina različitih distribucija, od kojih se većina po funkcionalnosti uopće ne razlikuje. Razlikuju se po dodatnim programima koji dolaze u paketu, a koji za korisnika nužno ne predstavljaju nikakvu dobit. Korisniku početniku ovo može biti dosta zbunjujuće, pa se preporučuje koristiti najpopularnija izdanja kao što su Ubuntu, Linux Mint, openSuse ili Arch Linux. Za ova distribucije korisnik može biti siguran da imaju adekvatnu podršku odnosno zajednicu koja redovito izdaje nove, poboljšane i sigurnije verzije. Osim velikog broja izdanja kao negativna strana navodi se i otežano snalaženje i korištenje novim korisnicima. Ovo je doista jedan subjektivan zaključak jer se u zadnjih desetak godina dosta vremena uložilo u samo grafičko sučelje i poboljšanje interakcije sa korisnikom. Danas je interakcija i jednostavnost korištenja na razini ostalih operacijskih sustava koji su dostupni na tržištu.

⁹ M. B. S. Zalavadia, *Linux Architecture and Advantages*, 2016.

Tablica 2: Sažetak prednosti i mana Linux operacijskih sustava

Prednosti	Mane
Besplatan	Previše distribucija
Sigurnost	Ograničena podrška određenih distribucija
Skalabilnost	Smanjen broj komercijalnih alata u odnosu na Windows platformu
Prenosivost	Nedostatak igara
Dizajniran za kontinuirani rad	
Velika zajednica	

(Prema: Ishan A B Ambanwela, 2014, An Introduction To Linux)

3.2.2. Arhitektura operacijskog sustava

Osnovna arhitektura Linux operacijskog sustava sastoji se od 5 slojeva:

- Jezgra
- Ljuska
- Korisničko sučelje
- Systemske biblioteke
- Programi sustava

Jezgra operacijskog sustava osnovni je građevni element operacijskog sustava koji sa jedne strane direktno komunicira sa sklopovljem računala, dok s druge strane komunicira sa ostatkom upravljačkih programa. Jezgra ima potpune ovlasti nad svim resursima računala, što znači da osim u iznimnim slučajevima kod prijenosa ovlasti, ostali računalni programi moraju preko jezgrinih funkcija pristupiti željenim resursima. Jezgra obavlja i ostale važne funkcije sustava kao što su upravljanje procesima, vremenom, prekidima, memorijom i sl.

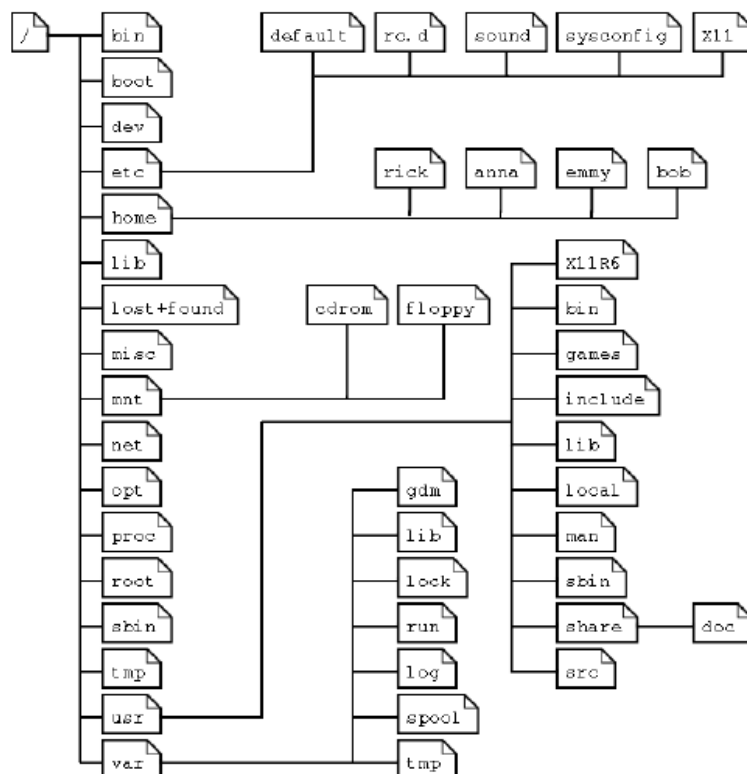
Odmah iznad jezgre nalazi se takozvana ljuska operacijskog sustava koja predstavlja korisničko sučelje prema operacijskom sustavu. Ona omogućava interakciju korisnika sa računalom. Interpretira jezgrene komande koje korisnik unosi i shodno tome izvršava određene zadatke.

Poznavatelji i stariji korisnici računalnih sustava općenito, dobro su upoznati sa radom u ljusci jer je ona predstavljala prvu aplikaciju putem koje je bilo moguće upravljati računalom. Naknadno su kreirana razna korisnička sučelja koja su omogućavala interaktivniji pristup računalu od suhoparnog upisivanja komandi. Korisnička sučelja predstavljaju vrlo intuitivan

način interakcije korisnika i operacijskog sustava jer omogućuju manipulaciju vizualnim elementima kao što su prozori, pokazivači, gumbi, kvadrati, obavijesti i sl. Razvoj naprednih grafičkih sučelja pratio je i razvoj perifernih uređaja kao što je i miš, bez kojega je danas interakcija sa računalom gotovo nezamisliva. Nadalje idući sloj arhitekture operacijskog sustava predstavljaju sistemske biblioteke koje sadrže specijalne funkcije, odnosno ponekad i čitave programe koje ostale, najčešće korisničke aplikacije koriste kako bi izvršile neku operaciju koja je pod kontrolom operacijskog sustava. Takve operacije nazivaju se sistemski pozivi. Zadnji sloj predstavljaju korisničke aplikacije odnosno programi sustava koji su odgovorni za obavljanje niza specijaliziranih zadataka.

3.2.3. Struktura direktorija

Na Linux operativnom sustavu sve se bazira na manipulaciji datotekama i direktorijima. Datoteke mogu predstavljati particije na disku ili čak sklopovske komponente. Kako bi način organizacije datoteka i direktorija bio unificiran, postoji standard pod imenom hijerarhija datotečnog sustava ili skraćeno FHS (*eng. Filesystem Hierarchy Standard*). Za jednostavnije upoznavanje sa osnovnom strukturom zadanom kroz FHS standard na Slika 3 prikazana je struktura, a dalje u tekstu biti će objašnjeni neki od osnovnih direktorija.



Slika 3: Struktura direktorija Linux operacijskog sustava (Dostupno na: <http://www.linuxandubuntu.com/home/the-linux-file-system-structure-explained>)

Tablica 3: Osnovni direktoriji na Linux operacijskom sustavu

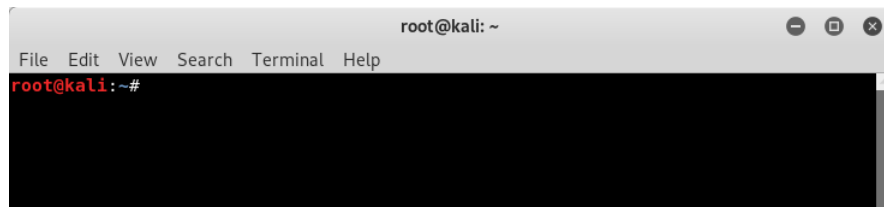
Direktorij	Primjena
/	Ishodišni direktorij prema FHS standardu. Svi ostali direktoriji na računalu su poddirektoriji ishodišnjog, ako se nalaze na istoj particiji
/bin	Sadrži niz izvršnih programa koji ostvaruju osnovne funkcije kod podizanja operacijskog sustava
/boot	Direktorij koji uključuje osnovne datoteke potrebne za podizanje Linux operacijskog sustava, uključujući i samu jezgru.
/dev	Sadrži upravljačke programe za sklopovlje računala
/etc	U pravilu sadrži osnovne konfiguracijske datoteke
/home	Ishodišni korisnički direktorij
/lib	Sadrži sistemske biblioteke
/root	Ishodišni direktorij za korisnika sa administratorskim ovlastima
/proc	Sadrži trenutno pokrenute jezgrene procese kao i podatke vezane uz memoriju računala
/tmp	Koristi se za pohranu privremenih podataka. Neki operacijski sustavi temeljeni na Linux-u periodički brišu nepotrebne podatke iz ovog direktorija
/sbin	Sadrži programe za administraciju sustava, oslanja se na izvršne datoteke iz /bin direktorija
/opt	Sadrži korisnički instalirane programske pakete
/var	Namijenjen za razne datoteke, najčešće zapise dnevnika

3.2.4. Osnovne naredbe

Upoznavanje sa svijetom Linux-a započeti će kroz upoznavanje sa nekima od osnovnih naredbi za rad u ljusci koje korisniku pojednostavljaju i ubrzavaju rad na ovoj platformi. Općenito je poznato da se sve naredbe u ljusci pišu malim slovima. Naime operativni sustav kod konvencije imenovanja razlikuje velika i mala slova, ali su za naredbe rezervirana isključivo mala.

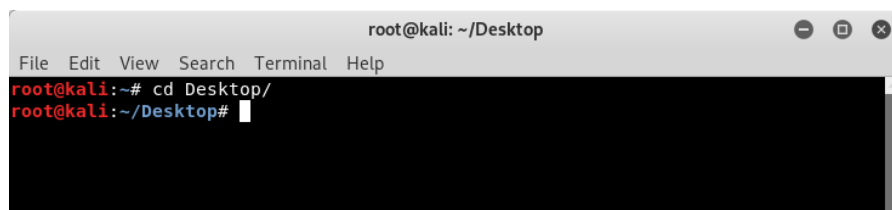
Uz same naredbe često se mora proslijediti i određeni argument. Argumenti su u pravilu opcije koje se mogu dodavati uz naredbe radi ostvarivanja složenijih operacija. Argumenti naredbe obično počinju sa znakom '-' nakon kojeg dolazi ime argumenta. Argumenti se mogu zadavati na dva načina, punim nazivom ili u skraćenoj verziji. Kod argumenata naredbe ne vrijedi pravilo da moraju počinjati malim slovom. Bitno je napomenuti da određene naredbe mogu primiti i više argumenata, pa se argumenti mogu ulančavati. Detaljniji prikaz korištenja naredbi i argumenata sa primjerima iz ljuske naći će se dalje u tekstu.

- **clear**
 - Naredba clear briše trenutni prikaz ekrana



Slika 4: Naredba clear

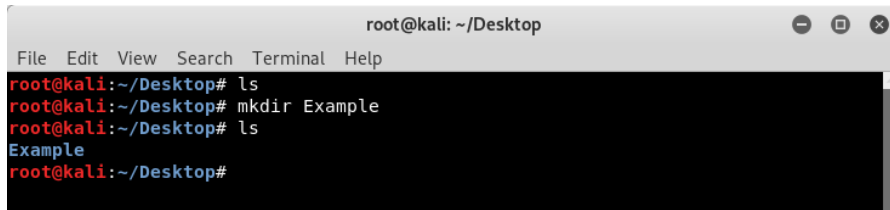
- **cd <putanja/do/direktorija>**
 - Naredba cd praćena putanjom pozicionira korisnika na odabrani direktorij



Slika 5: Naredba cd

- **mkdir <ime_direktorija>**

- Naredba mkdir kreira direktorij sa zadanim imenom

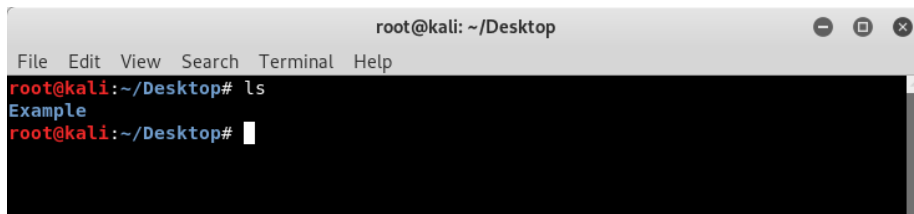


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls
root@kali:~/Desktop# mkdir Example
root@kali:~/Desktop# ls
Example
root@kali:~/Desktop#
```

Slika 6: Naredba mkdir

- **ls**

- Naredba ls ispisuje sadržaj direktorija u kojemu se trenutno nalazi korisnik. Ispisuje poddirektorije i datoteke u odabranom direktoriju

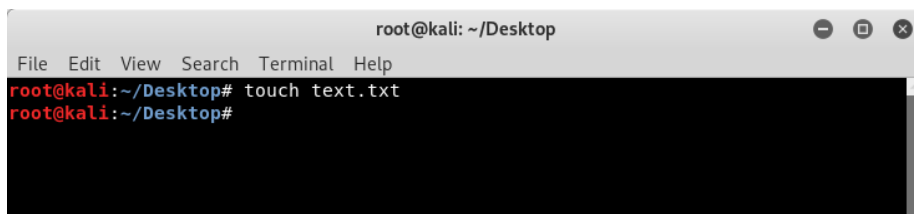


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls
Example
root@kali:~/Desktop#
```

Slika 7: Naredba ls

- **touch <ime_datoteke>**

- Naredba touch kreira datoteku sa željenim imenom

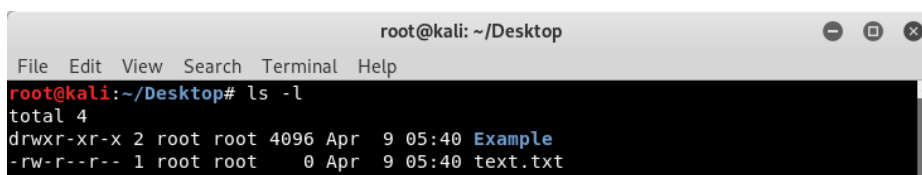


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# touch text.txt
root@kali:~/Desktop#
```

Slika 8: Naredba touch

- **ls -l**

- Naredba ls -l ispisuje sadržaj direktorija u obliku liste sa dodatnim svojstvima direktorija i datoteka

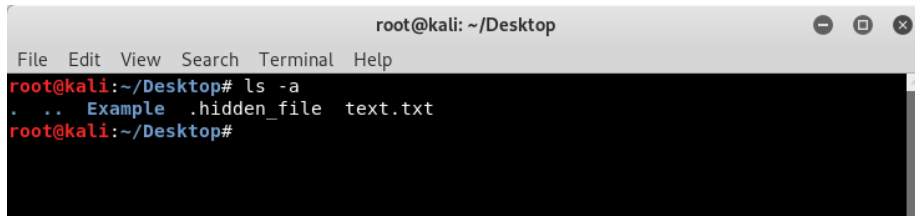


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls -l
total 4
drwxr-xr-x 2 root root 4096 Apr  9 05:40 Example
-rw-r--r-- 1 root root    0 Apr  9 05:40 text.txt
root@kali:~/Desktop#
```

Slika 9: Naredba ls -l

- **ls -a**

- Naredba ls -a ispisuje sadržaj direktorija sa skrivenim datotekama i poddirektorijima

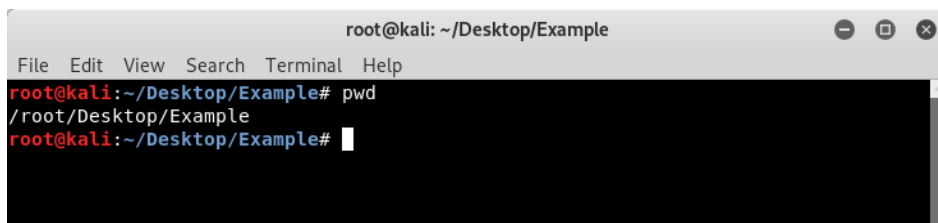


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls -a
. .. Example .hidden_file text.txt
root@kali:~/Desktop#
```

Slika 10: Naredba ls -a

- **pwd**

- Naredba pwd ispisuje putanju do direktorija u kojem se korisnik trenutno nalazi

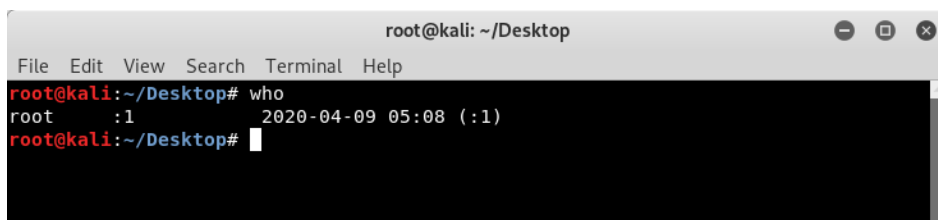


```
root@kali: ~/Desktop/Example
File Edit View Search Terminal Help
root@kali:~/Desktop/Example# pwd
/root/Desktop/Example
root@kali:~/Desktop/Example#
```

Slika 11: Naredba pwd

- **who**

- Naredba who ispisuje trenutno prijavljenog korisnika sa dodatnim informacijama kao što su vrijeme prijave ili adresa sa koje se korisnik prijavio

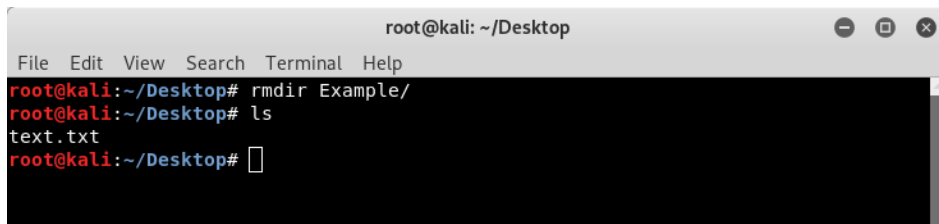


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# who
root :1 2020-04-09 05:08 (:1)
root@kali:~/Desktop#
```

Slika 12: Naredba who

- **rmdir <ime_direktorija>**

- Naredba rmdir briše odabrani direktorij pod uvjetom da je isti prazan

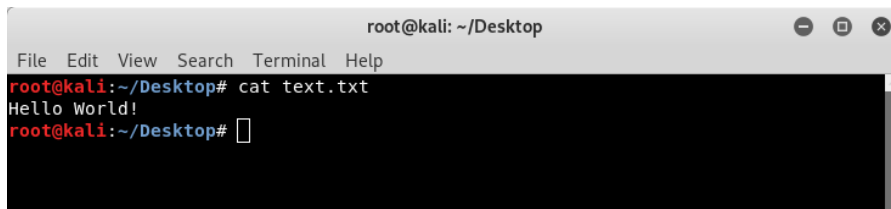


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# rmdir Example/
root@kali:~/Desktop# ls
text.txt
root@kali:~/Desktop#
```

Slika 13: Naredba rmdir

- **cat <ime_datoteke>**

- Naredba cat ispisuje sadržaj odabrane datoteke

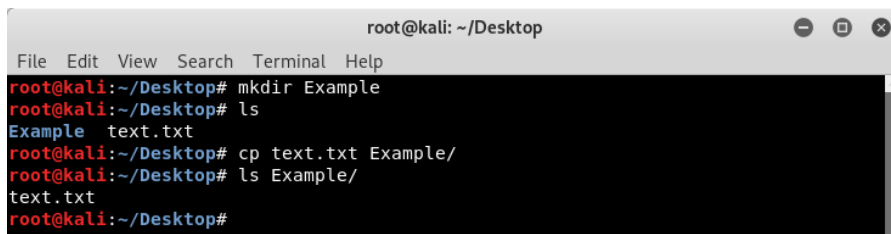


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# cat text.txt
Hello World!
root@kali:~/Desktop#
```

Slika 14: Naredba cat

- **cp <putanja/sa/koje/kopiramo> <putanja/na/koju/kopiramo>**

- Naredba cp kopira podatke sa ishodišne na odredišnu putanju

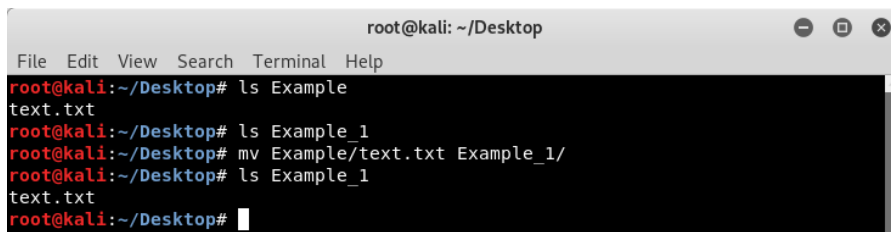


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# mkdir Example
root@kali:~/Desktop# ls
Example text.txt
root@kali:~/Desktop# cp text.txt Example/
root@kali:~/Desktop# ls Example/
text.txt
root@kali:~/Desktop#
```

Slika 15: Naredba cp

- **mv <putanja/sa/koje/premještamo> <putanja/na/koju/premještamo>**

- Naredba mv premješta podatke sa ishodišne na odredišnu putanju

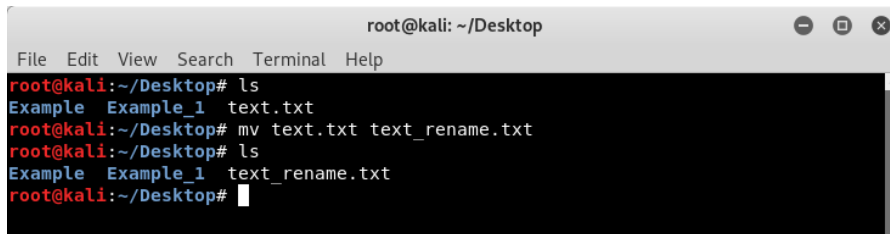


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls Example
text.txt
root@kali:~/Desktop# ls Example_1
root@kali:~/Desktop# mv Example/text.txt Example_1/
root@kali:~/Desktop# ls Example_1
text.txt
root@kali:~/Desktop#
```

Slika 16: Naredba mv - premještanje

- **mv <staro_ime> <ново_ime>**

- Naredba mv na Linux platformi koristi se kao operator preimenovanja u slučaju da se umjesto putanja navede staro i novo ime direktorija ili datoteke

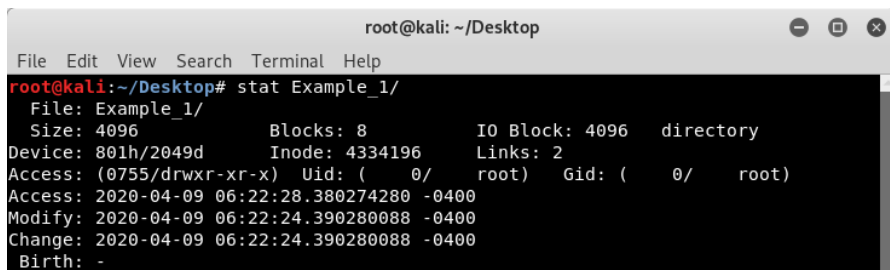


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# ls
Example Example_1 text.txt
root@kali:~/Desktop# mv text.txt text_rename.txt
root@kali:~/Desktop# ls
Example Example_1 text_rename.txt
root@kali:~/Desktop#
```

Slika 17: Naredba mv - preimenovanje

- **stat <ime_datoteke>**

- Naredba stat ispisuje detaljne informacije o datoteci ili direktoriju

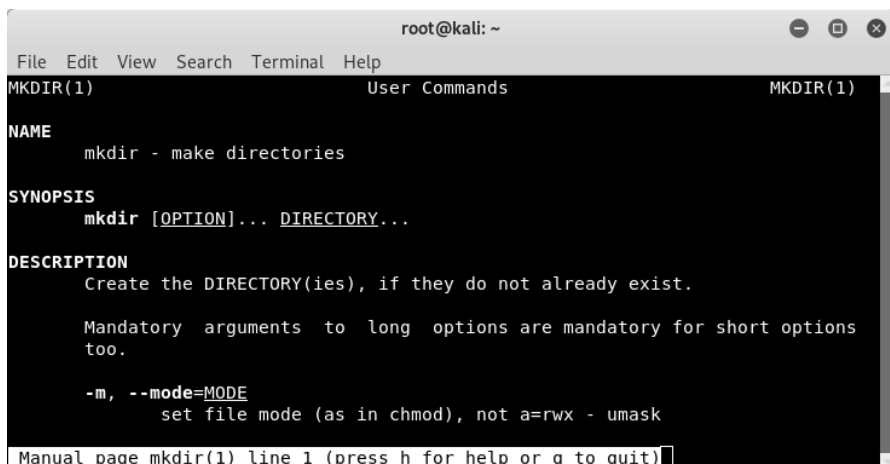


```
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# stat Example_1/
  File: Example_1/
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 801h/2049d  Inode: 4334196    Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2020-04-09 06:22:28.380274280 -0400
Modify: 2020-04-09 06:22:24.390280088 -0400
Change: 2020-04-09 06:22:24.390280088 -0400
 Birth: -
```

Slika 18: Naredba stat

- **man <naredba>**

- Naredba man koristi se kao pomoć pri korištenju naredbi. Za zadanu naredbu daje opis korištenja sa svim mogućim argumentima koje naredba može primiti



```
root@kali: ~
File Edit View Search Terminal Help
MKDIR(1) User Commands MKDIR(1)
NAME
  mkdir - make directories
SYNOPSIS
  mkdir [OPTION]... DIRECTORY...
DESCRIPTION
  Create the DIRECTORY(ies), if they do not already exist.

  Mandatory arguments to long options are mandatory for short options
  too.

  -m, --mode=MODE
      set file mode (as in chmod), not a=rwx - umask
Manual page mkdir(1) line 1 (press h for help or q to quit)
```

Slika 19: Naredba man

Osim ovih jednostavnih naredbi na Linux platformi postoji i cijeli set složenijih koje nisu obuhvaćene ovim radom. Složenije naredbe zahtijevaju dublje razumijevanje sustava, stoga će se jednostavno preskočiti. Ova razina razumijevanja ljsuke, uz određena znanja osnovnih koncepata programskih jezika biti će dovoljna za uspješno savladavanje programskog rješenja iz ovoga rada.¹⁰

3.3. Udaljeni pristup

Kada korisnik nema fizički pristup računalu ili u slučaju Raspberry Pi računala nema dodatni monitor na koji će spojiti uređaj, na neki način korisniku se mora omogućiti udaljeni pristup. Upravljanje računalom na daljinu u pravilu se može izvoditi na dva načina: pristupom ljsuci ili pristupom grafičkom sučelju odnosno radnoj površini računala. Osnove udaljenog pristupa nalažu da postoje dvije strane: poslužiteljska i klijentska. Poslužiteljska dozvoljava pristup klijentskoj strani koristeći se jednim od mrežnih protokola koji omogućavaju komunikaciju računala na mreži.

Među najpoznatijima protokolima sigurno su: SSH (*eng. Secure shell*) odnosno protokol za ostvarivanje sigurne komunikacije, RDP (*eng. Remote Desktop Protocol*) odnosno protokol udaljenog pristupa radnoj površini računala, SFTP (*eng. Secure File Transfer Protocol*) protokol sigurnog prijenosa podataka, FTP (*eng. File Transfer Protocol*) protokol prijenosa podataka i drugi. U nastavku poglavlja proći će se kroz osnove mrežnih tehnologija, mogućnosti spajanja i osnovnih koncepata nekih od nabrojanih protokola.

3.3.1. SSH protokol

SSH protokol razvijen je sa namjerom osiguravanja tajnosti odnosno mogućnosti kriptiranja podataka koji se prenose po računalnoj mreži. Danas predstavlja zlatni standard u sigurnoj mrežnoj komunikaciji. Zamijenio je Telnet protokol čija je funkcija također bila udaljeni pristup računalima na razini ljsuke ili komandne linije, bez mogućnosti ostvarivanja sigurne konekcije. Za razliku od nesigurnih protokola kod kojih je korisnik u mogućnosti presresti, pročitati i promijeniti podatke u toku slanja, kod SSH protokola svi podaci su prije slanja kriptirani te kao takvi putuju mrežom do odredišta gdje se u konačnici dekriptiraju i postaju jasno razumljivi korisniku. Za kriptiranje najčešće se koriste simetrični algoritmi kriptiranja kao

¹⁰ M. Garrels, *Introduction to Linux*, 2008.

što su AES (*eng. Advanced Encryption Standard*), DES (*eng. Data Encryption Standard*), 3DES (*eng. Triple Data Encryption Standard*) i ostali. Kod simetričnih algoritama kriptiranja matematičkom funkcijom po određenom ključu ulazni podaci pretvaraju se u određeni niz. Bez poznavanja odabranog ključa gotovo je nemoguće otkriti izvorne podatke. Kod simetričnih algoritama javlja se problem komuniciranja tajnog ključa sa stranom primatelja. Za razmjenu tajnog ključa koriste se drugi algoritmi temeljeni na matematičkim operacijama potenciranja i logaritmiranja i u ovom radu neće se detaljnije opisivati. Bitno je naglasiti samo da je gotovo nemoguće u realnom vremenu „razbiti“ odnosno pronaći tajni ključ obzirom na poznate performanse današnjih računala.

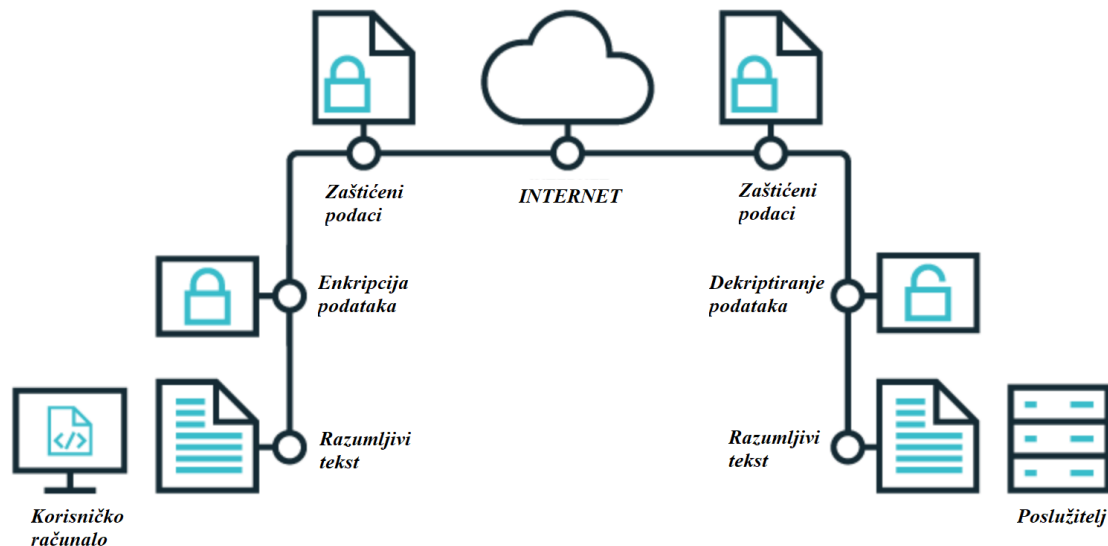
Osim simetričnog načina kriptiranja kod SSH protokola koristi se i asimetrično kriptiranje kod kojeg se pak koriste parovi privatnih i javnih ključeva. Javni ključ u ovom slučaju poznat je svim stranama u komunikaciji dok je privatni ključ kao što mu i samo ime govori poznat samo ovlaštenim stranama. Najpoznatiji asimetrični algoritmi kriptiranja koji se koriste kod SSH protokola su DSA (*eng. Digital Signature Algorithm*) i RSA (*eng. Rivest – Shamir - Adleman*) algoritmi. Treba naglasiti da su asimetrični algoritmi puno složeniji i koriste se u kriptiranju manjih količina podataka. SSH protokol omogućio je i unapređivanje ostalih mrežnih protokola. Kroz neke od njih proći će se u daljnjem tekstu, a najbitniji od njih su SCP (*eng. Secure Copy Protocol*) koji omogućuje sigurno kopiranje podataka sa jednog na drugo računalo u mreži, SSHFS (*eng. Secure Shell File System*) protokol koji omogućuje manipulaciju datotečnim sustavom udaljenog računala, te SFTP (*eng. Secure File Transfer Protocol*) koji definira sigurnu razmjenu podataka između računala.¹¹

3.3.2. SFTP protokol

SFTP (*eng. Secure File Transfer Protocol*) je mrežni protokol koji omogućuje siguran pristup, razmjenu i manipulaciju podacima na udaljenom računalu putem SSH konekcije. Nastao je kao poboljšanje FTP (*eng. File Transfer Protocol*) protokola od kojeg je naslijedio osnove, ali se od njega bitno razlikuje. SFTP protokol koristi sigurni kanal za prijenos kriptiranih podataka, za razliku od FTP protokola kod kojeg je prijenos zaštićen lozinkom, koja se kao i podaci šalju mrežom u obliku teksta. Na taj način zlonamjerna korisnik na mreži može na jednostavan način presretanjem poruke doći do njezinog sadržaja.

¹¹ D. J. Barrett, R. E. Silverman, *SSH, the secure shell: the definitive guide*, 2001.

Obzirom na način implementacije prijenosa podataka kroz mrežu na Windows i Linux platformama korištenje SFTP protokola omogućeno je putem aplikacijskih rješenja. Jedna od najpoznatijih su WinSCP na Windows platformi, odnosno FileZilla na Linux platformi. ¹²



Slika 20: Prijenos podataka SFTP protokolom (prema: John Hughes, *An introduction to Secure Shell Access and Secure File Transfer Protocol*, 2018)

3.3.3. SCP protokol

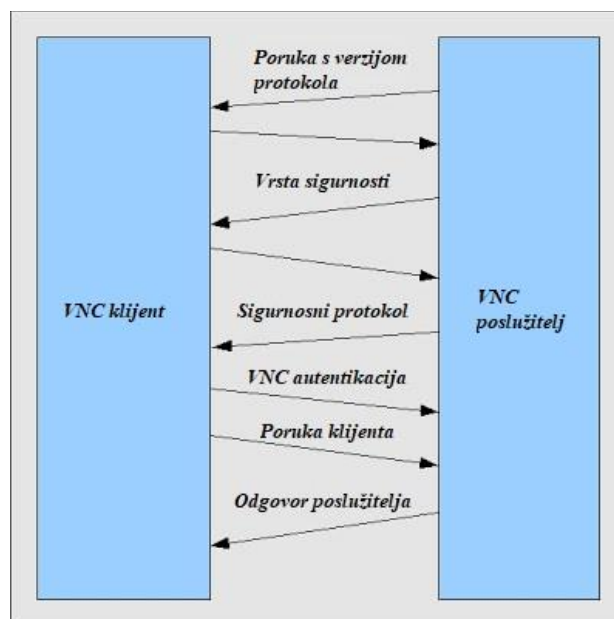
SCP (eng. *Secure Copy Protocol*) gotovo je identičan kao i SFTP protokol. I on koristi SSH kao osnovu za uspostavljanje sigurne konekcije i prijenosa kriptiranih podataka. Od SFTP protokola razlikuje se u načinu implementacije algoritama za prijenos podataka. Način implementacije algoritama diktira i samu brzinu, pa tako može se reći da je prijenos podataka SCP protokolom brži od prijenosa SFTP, zbog toga što se ne čeka potvrda dostave paketa. Naravno ova značajka direktno utječe na neke druge funkcionalnosti kao što je interaktivnost pri korištenju. Naime kod SCP protokola prekidom prijenosa, isti nije moguće nastaviti već je sa prijenosom potrebno krenuti iz početka. Prijenos SCP protokolom ograničen je na korištenje komandne linije. ¹³

¹² C. Chung, *SSH File Transfer Protocol (SFTP) Explained*, 2014.

¹³ Marc Wilson, *SCP – What is Secure Copy Protocol – Definition & Example*, 2019. <https://www.pcwld.com/what-is-scp> (dostupno 21.04.2020).

3.3.4.VNC protokol

VNC (*eng. Virtual Network Computing*) odnosno protokol za virtualno računalno umrežavanje omogućuje povezivanje s udaljenim računalom i kontrolu zaslona. Obzirom da Raspberry Pi računalo ne posjeduje ugrađeni zaslon mnogi korisnici posegnuti će za ovim vrlo jednostavnim, a efikasnim rješenjem. VNC nudi potpunu razinu interakcije kao i kod klasičnog korištenja računala. Implementacija ovog protokola u većini slučajeva kao podlogu koristi SSH protokol za ostvarivanje sigurne komunikacije i prijenosa podataka. Mrežom se šalju kriptirane korisničke akcije i položaj grupe piksela koji služe za iscrtavanje zaslona poslužiteljskog računala.¹⁴

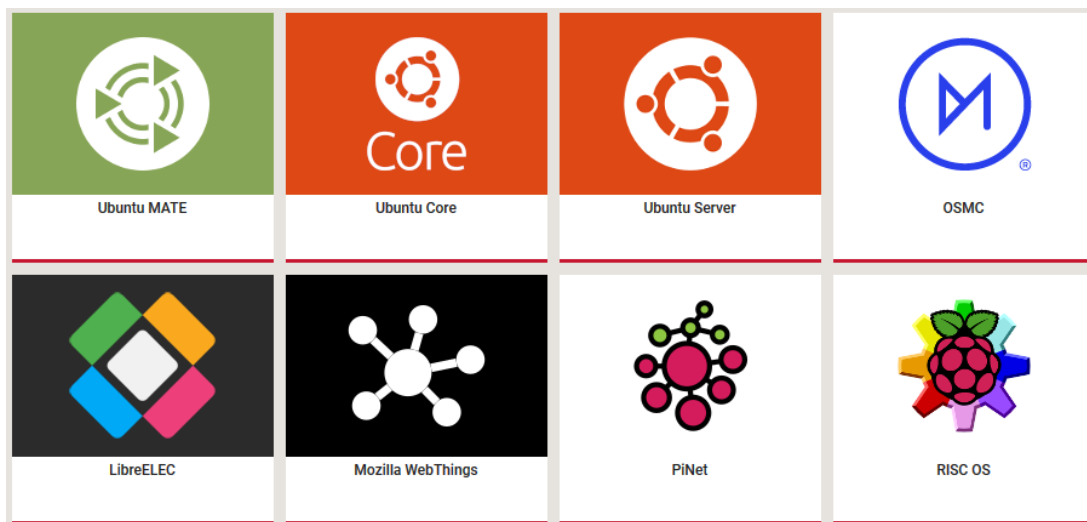


Slika 21: Komunikacija VNC protokolom

¹⁴ VNC® User Guide. RealVNC Limited, 2015.

4. Operacijski sustavi

Operacijski sustav osnovni je alat odnosno skup programa koji omogućuju komunikaciju sa sklopovljem računala. Zadužen je za upravljanje svim računalnim komponentama i skupa s njima čini cjelinu. Ta cjelina sastoji se od osnovnih dijelova kao što su jezgra operacijskog sustava, ljuska, datotečni sustav i korisničko sučelje. Svaka od cjelina ima za zadaću efikasno iskorištavanje računalnih resursa izvođenjem određenih funkcija kao što su alokacija memorije, upravljanje rasporedom izvođenja programa, upravljanjem procesima i njihovim prekidima, te upravljanje datotečnim sustavom. Nadalje osnovnu svrhu ispunjava pružajući korisniku mogućnosti unosa podataka u sustav, manipulaciju podacima u sustavu, te prikaz rezultata obrade nad podacima. Osim izvođenja osnovnih funkcija svaki operacijski sustav moguće je proširiti dodatnim korisničkim programima kojima je zadaća podizanje efikasnosti u izvođenju složenijih funkcija.¹⁵



Slika 22: Operacijski sustavi za Raspberry Pi (Dostupno na:

<https://www.raspberrypi.org/downloads/>)

Operacijske sustave u pravilu dijele se prema korisničkom sučelju, prema broju zadataka koje mogu obavljati i prema broju korisnika. Korisničko sučelje na starijim operacijskim sustavima svodilo se na prikaz komandne linije i upisivanje naredbi. Podjela procesorskog

¹⁵ J. Leonardo, K. Milan, C. Dario, *Sustavska programska potpora*, 2010.

vremena i njegovo optimalno iskorištavanje dovelo je do razvoja modernih grafičkih korisničkih sučelja, gdje korisnik pritiskom na tipku miša zadaje određene akcije računalu.

Prema broju korisnika mogu se razlikovati jednokorisnički ili višekorisnički operacijski sustavi. Jednokorisnički operacijski sustavi su najčešće sustavi za osobna računala koji dozvoljavaju samo jednom korisniku da u određenom trenutku pristupa podacima. U velikoj mjeri takvi operacijski sustavi imaju mogućnosti stvaranja profila sa osobnim memorijskim prostorom kako bi više korisnika moglo koristiti isto računalo. Višekorisnički operacijski sustavi suprotno jednokorisničkim dozvoljavaju pristup većem broju korisnika koji na istom sustavu mogu u isto vrijeme obavljati različite funkcije. Ovakav način korištenja sustava najčešće se provodi udaljenim pristupom kada se korisnici putem komandne linije i korištenjem određenih mrežnih protokola spajaju na centralizirano računalo odnosno poslužitelj. Poslužitelj je u stanju obrađivati zahtjeve od više korisnika istovremeno. Prema broju zadataka postoje operacijski sustavi koji istovremeno mogu obavljati više procesa ili samo jedan proces. Operacijski sustavi koji omogućavaju istovremeno obavljanje više procesa rade to na dva načina: podjelom procesorskog vremena ili pred definiranim stanjima procesa koji „prosljeđuju“ kontrolu nad procesorom.

Većina operacijskih sustava za Raspberry Pi računalo temelji se na Linux jezgri. To znači da omogućavaju korisniku rad iz komandne linije ili sa grafičkog sučelja. Također omogućuju višekorisnički rad i obavljanje više procesa od jednom. U pravilu Linux platforma omogućuje i objedinjuje sve najbolje od modernih operacijskih sustava danas. Osim operacijskih sustava temeljenih na Linux jezgri, Raspberry Pi računala u mogućnosti su pokretati i druge specijalizirane operacijske sustave kao na primjer RiscOS ili Windows IoT operacijske sustave. Ukratko za RiscOS operacijski sustav bitno je znati da je namijenjen za računala temeljena isključivo na ARM procesorskoj arhitekturi i na tržištu postoji već nekih tridesetak godina, dok je Windows IoT nešto novijeg datuma i namijenjen je razvoju interneta stvari i povezivanja raznih senzora korištenjem Windows platforme. Kroz iduća poglavlja detaljnije će se kroz gore navedene operacijske sustave za Raspberry Pi kako bi se stekao dojam o kompleksnosti korištenja, te ukazalo na potencijalne prednosti i mane svakog od operacijskih sustava.

4.1. Raspbian

Prvi i službeni operacijski sustav za Raspberry Pi dolazi u inačici pod imenom Raspbian. Ovaj operacijski sustav temelji se na Linux jezgri i ovisno o paketu dolazi sa pred instaliranim korisničkim programima za edukaciju, programiranje i opću upotrebu. Korisniku pruža mogućnost rada iz komandne linije ili interakciju putem grafičkog sučelja. Na tržištu postoje tri verzije Raspbian operacijskog sustava:¹⁶

- Raspbian Buster Lite
- Raspbian with desktop
- Raspbian with desktop and recommended software



Slika 23: Radna površina Raspbian operacijskog sustava

4.1.1. Instalacija operacijskog sustava

Instalacija Raspbian operacijskog sustava započinje odabirom željene verzije. Prva i veličinom najmanja verzija pod imenom „Raspbian Buster Lite“ korisniku omogućava rad isključivo putem komandne linije. Druge dvije verzije dolaze sa grafičkim sučeljem, a razlikuju

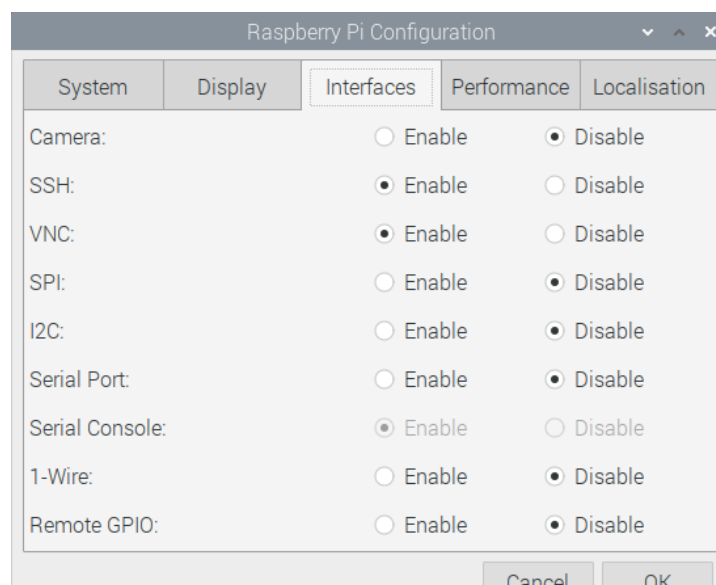
¹⁶ Raspbian, 2020. <https://www.raspberrypi.org/documentation/raspbian/> (dostupno 21.04.2020.)

se po količini korisničkih alata koji se nalaze u instalacijskom paketu. Kako Raspberry Pi ne posjeduje ugrađenu memoriju u obliku tvrdog diska, operacijski sustav odnosno njegov paket potrebno je putem specijalnog alata instalirati na memorijsku karticu. Alat za instalaciju univerzalan je, te se može koristiti i za instalaciju ostalih operacijskih sustava. Nakon umetanja kartice na kojoj se nalazi operacijski sustav u Raspberry Pi i paljenja računala, instalacija operacijskog sustava počinje automatski. Korisniku je interakcija sa računalom tokom instalacije vrlo ograničena.

Ne dopuštaju se napredne mogućnosti kao što su formatiranje ili particioniranje memorije. Omogućene su osnovne funkcije kao na primjer postavljanje lokalizacije sustava, spajanje na mrežu i preuzimanje nadogradnji. Niz instrukcija vodi korisnika ka postavljanju funkcionalnog sustava.

4.1.2. Konfiguracija sustava

Po završetku instalacije korisnik bi trebao postaviti osnovnu konfiguraciju sustava. U postavkama sustava preporučuje se promijeniti lozinku, zatim odabrati da li će se korisnika automatski prijavljivati u sustav nakon paljenja računala. Korisnik na zaslonu sa sučeljima može na vrlo jednostavan način omogućiti ili onemogućiti korištenje određenih sklopovskih elemenata kao što su sučelje za serijsku komunikaciju, ulazi izlazi opće namjene ili određenih protokola za udaljeni pristup kao što su SSH, odnosno VNC o kojima smo govorili u ranijim poglavljima. Ostale konfiguracijske mogućnosti ne razlikuju se od klasičnih na Linux okruženjima.



Slika 24: Konfiguracija sučelja na Raspbian operacijskom sučelju

4.1.3. Korisnički programi

Instalacijom verzije u koju su uključeni svi preporučeni korisnički programi korisnik u glavnom izborniku ima mogućnost odabira programa iz čak osam kategorija: programiranje, edukacija, ured, Internet, zvuk i video, grafika, igre i pomagala. Prva kategorija „Programiranje“ sadrži paket alata odnosno integriranih razvojnih okruženja za razne programske jezike. Za sada je sigurno da Raspbian standardno podržava Python i Java programske jezike. Detaljnije o programskim jezicima na Raspberry Pi računalu može se naći u zasebnom poglavlju. Kategorija „Edukacija“ sadrži alat za simuliranje elektroničkih digitalnih sklopova, dok kategorija „Ured“ nudi set alata za obradu teksta i slike istovjetan Microsoft Office paketu na Windows platformi. Kategorija „Internet“ osim web preglednika sadrži i mail klijenta te program za korištenje VNC protokola. Kategorije „Grafika“ i „Igre“ sadrže alate za reprodukciju audio i video zapisa, te igre razvijene specijalno za Raspberry Pi. Možda i najvažniji set alata nalazi se u kategoriji „Pomagala“. Neki od alata služe za upravljanje datotekama i procesima, otvaranju komandne linije ili pokretanju jednostavnog uređivača teksta.

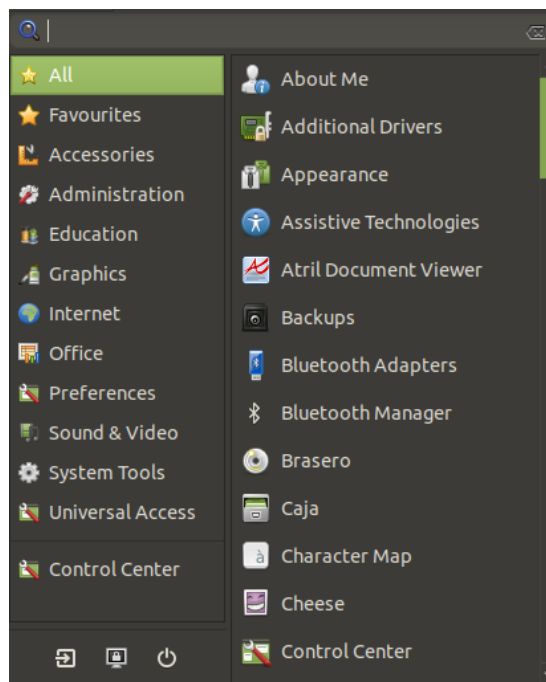
4.1.4. Značajke operacijskog sustava

U prvom redu treba navesti jednostavnost instalacije samog operacijskog sustava. Korisnici koji su ikada probali instalaciju operacijskog sustava na klasičnom računalu, neovisno o platformi koju instaliraju, upoznati su sa nizom postavki koje je potrebno podesiti tijekom instalacije. Nadalje poznato je da je Raspbian operacijski sustav kompatibilan sa svim Raspberry Pi računalima.

4.2. Ubuntu MATE

Ubuntu MATE poznatiji je operacijski sustav iz Linux zajednice zbog svoje inkarnacije sa stolnih računala. Ovaj popularni operacijski sustav svoju je verziju ponudio i za Raspberry Pi platformu, koja izgledom nimalo ne odstupa od klasične desktop izvedbe. Ovaj operacijski sustav ne poznaje podjelu prema razini paketa kao Raspbian, ali dolazi u dvije inačice. Jedna inačica koristi se za 32 bitnu procesorsku arhitekturu dok je druga namijenjena 64 bitnoj procesorskoj arhitekturi. ¹⁷

¹⁷ Ubuntu MATE for Raspberry Pi, 2020. <https://ubuntu-mate.org/ports/raspberry-pi/> (dostupno 21.04.2020.)



Slika 25: Struktura programskih paketa Ubuntu MATE operacijskog sustava

4.2.1. Instalacija operacijskog sustava

Instalacija Ubuntu MATE operacijskog sustava na Raspberry Pi gotovo je istovjetna instalaciji verzije za klasično stolno ili prijenosno računalo. Iz ove perspektive daleko je kompliciranija od instalacije Raspberry Pi računala. Iako se kao i kod Raspbian-a ne dopušta formatiranje i particioniranje memorije, od korisnika se zahtijevaju druge konfiguracijske akcije. Većinu konfiguracijskih koraka Raspbian operacijski sustav odradi u pozadini ili ostavi korisniku na volju da kasnije mijenja konfiguracijske parametre, dok Ubuntu MATE izričito traži postavljanje istih tijekom instalacije.

4.2.2. Konfiguracija sustava

Kako smo već naveli da se većina konfiguracijskih akcija odvija tijekom same instalacije operacijskog sustava, korisniku ne preostaje ništa nego konfiguracija naprednijih postavki sustava kao i u svakom drugom Linux okruženju, a kroz koje se sada neće detaljnije prolaziti.

4.2.3. Korisnički programi

Glavni izbornik Ubuntu MATE operacijskog sustava otkriva nam hijerarhiju od dvanaest kategorija u koje su svrstani korisnički programi. Korisnici raznih distribucija Linuxa ovdje neće naći neke novosti, dok će se novi korisnici definitivno teže snalaziti nego na Raspbian operacijskom sustavu. Većina programa jednaka je kao i na Raspbian operacijskom sustavu, ali može se primijetiti da je ovaj operacijski sustav namijenjen više općem korištenju računala jer ne dolazi sa nikakvim pred instaliranim programima za edukaciju. Osnovni paket u sebi ipak sadrži podršku za Python programski jezik, dok će pobornici Java programskog jezika biti primorani prvo instalirati Java virtualni stroj, a po želji i neko od razvojnih okruženja.

4.2.4. Značajke operacijskog sustava

Velika prednost Ubuntu MATE operacijskog sustava je direkta veza sa Ubuntu Linux zajednicom, što garantira redovite sistemske nadogradnje. Na taj način korisnik može biti siguran da su programski paketi na najvećoj sigurnosnoj ali i operativnoj razini. Obzirom da je baziran na Ubuntu inačici ima pristup svim Ubuntu programskim paketima. Korisnik tako ima gotovo neograničene mogućnosti proširivanja funkcionalnosti računala. Upravljanje ulazno izlaznim sučeljem opće namjene ovdje se također podrazumijeva.

4.3. Ubuntu Server

Kao što mu i samo ime kaže Ubuntu Server poslužiteljski je oblik operacijskog sustava. Omogućuje korisnicima od klasičnog dijeljenja sadržaja preko mreže, korištenja naprednih virtualizacijskih alata, korištenja drugih poslužiteljskih servisa za razvoj web aplikacija, pa sve do najnovijih tehnologija kao što su računanje u oblaku. Već iz priloženog može se zaključiti da ovaj operacijski sustav nije za svakoga, već da će ga ponajprije koristiti iskusniji korisnici koji će moći iskoristiti njegov puni potencijal.¹⁸

¹⁸ Ubuntu Server for ARM, 2020. <https://ubuntu.com/download/server/arm> (dostupno 21.04.2020.)

4.3.1. Instalacija operacijskog sustava

Sama instalacija na Raspberry Pi platformi u potpunosti je automatizirana i od korisnika ne zahtjeva gotovo nikakvu interakciju. Po završetku instalacije jedina akcija koju korisnik mora izvršiti jest promjena lozinke za prijavu u sustav. Bitno je naglasiti da ovo nije operacijski sustav sa grafičkim sučeljem pa se sva interakcija obavlja putem komandne linije.

4.3.2. Konfiguracija sustava

Na Ubuntu Server operacijskom sustavu sva konfiguracija sustava odvija se manipulacijom parametara u konfiguracijskim datotekama. Omogućuje potpunu kontrolu nad svim parametrima sustava kao i na bilo kojem drugom Linux okruženju. Bitno je naglasiti da konfiguracija ovakvog sustava nije nimalo jednostavna jer zahtijeva dublje znanje o arhitekturi Linux-a i njegovog datotečnog sustava.

4.3.3. Korisnički programi

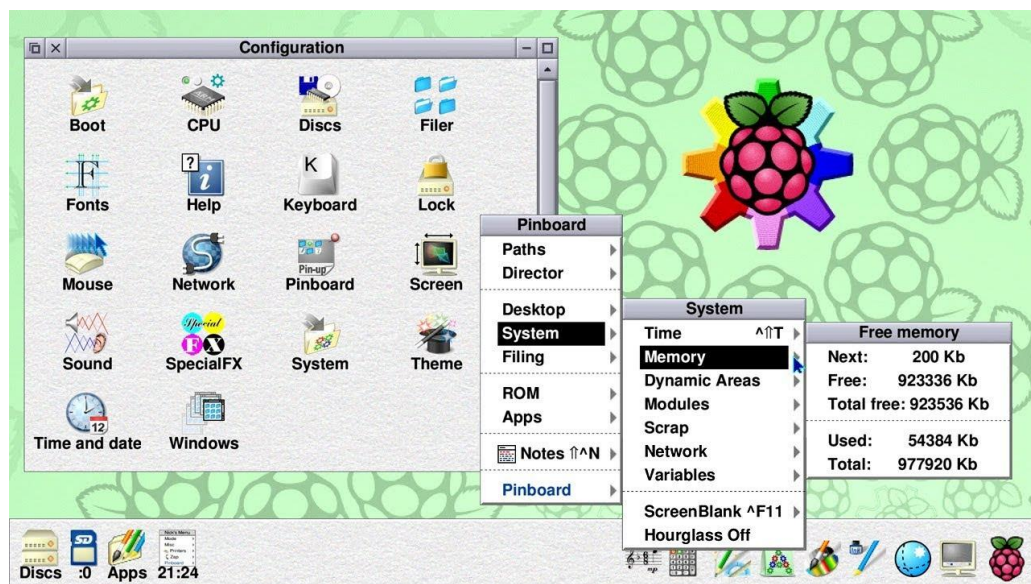
Osim standardnog paketa Linux korisničkih programa za komandnu liniju koji dolaze u svim verzijama ove platforme, na ovom operacijskom sustavu neće se pronaći ostali korisnički programi. Može se reći da podržava Python programski jezik, dok će zaljubljenici u Javu isto kao i na Ubuntu MATE operacijskom sustavu morati naknadno instalirati sve programske pakete.

4.3.4. Značajke operacijskog sustava

Glavna prednost ovog operacijskog sustava je zapravo činjenica da je besplatan. Bez uloženog novca i malo više znanja na vrlo jednostavan način moguće je podignuti vlastiti poslužitelj sa svim servisima poput poslužitelja baza podataka, web poslužitelja ili običnih skladišta podataka na koje bi ste inače trošili novce plaćajući usluge u oblaku. Mogućnosti personalizacije su gotovo beskonačne, a ovise o mašti i količini znanja korisnika. Osim toga zbog iznimno malih sklopovskih zahtjeva od 512MB za radnu memoriju, 1GHz za procesor i 2GB memorijskog prostora moguće ga je instalirati na jeftine uređaje koji će se uslijed kvara sa minimalnim troškovima bez problema zamijeniti.

4.4. RiscOS

Iako stariji od Linux operacijskih sustava RiscOS ostao je negdje zaboravljen, ponajprije zbog činjenice da je od početka bio operacijski sustav zatvorenog koda, a svoju primjenu nalazio je u velikim industrijskim postrojenjima gdje je automatizirao procese koristeći računala sa ARM procesorskom arhitekturom. Osim što u počecima nije bio dostupan na osobnim računalima, još jedna boljka mu je sporo izdavanje nadogradnji. Zakonom tržišta bolje organizirani timovi na jednostavan su način preskočili ovaj operacijski sustav bržom prilagodbom zahtjevima tržišta. Tek nakon dvadesetak godina ovaj operacijski sustav dobio je inačicu pod imenom „RISC OS Open Limited“ koja je djelomično podijelila dio koda sa zajednicom i dala ga na korištenje odnosno omogućila izradu samostalnih programskih rješenja.¹⁹



Slika 26: Radna površina RISC operacijskog sustava (Preuzeteo sa:

<https://i.ytimg.com/vi/oL4w3AK6Qpw/maxresdefault.jpg>)

4.4.1. Instalacija operacijskog sustava

Ako je suditi po ostalim do sada navedenim operacijskim sustavima, instalacija RiscOS-a je definitivno najjednostavnija. Pri podizanju operacijskog korisnik nema nikakvu interakciju s računalom. Za nekoliko sekundi operacijski sustav je u potpunosti instaliran, a pred korisnikom se nalazi radna površina. U ovom trenutku korisnik je spreman za rad na računalu.

4.4.2. Konfiguracija sustava

RiscOS nudi korisnicima jednostavno korisničko sučelje nešto poput onoga na Windows ili Linux okruženjima sa grafičkom podrškom. Korisnici su u mogućnosti konfigurirati osnovne parametre sustava vezane uz audio vizualne komponente, priključenu periferiju ili pak mrežu. Od ostalih operacijskih sustava razlikuje se po mogućnosti konfiguracije parametara vezanih uz sami procesor, odnosno način na koji procesor izvršava određene instrukcije.

4.4.3. Korisnički programi

Obzirom na relativnu malu zajednicu ovaj operacijski sustav raspolaže zavidnom količinom korisničkih programa. Dio korisničkih programa kao što su uređivači teksta, programi za audio i video reprodukciju ili web preglednik dolaze pred instalirani dok su ostali dostupni kroz upravitelj paketa. U upravitelju paketa pronaći će se dovoljno korisničkih programa za podršku osnovnom radu računala, dok će korisnici ostati razočarani već ako potraže nešto poput programa za proračunske tablice. Treba naglasiti da sa većinom programskih paketa na Raspberry Pi postoje određeni problemi, pa je korisničko iskustvo u najmanju ruku rečeno u velikoj mjeri narušeno. Ljubitelji razvoja modernih programskih rješenja ostati će isto tako razočarani izborom podržanih jezika. Ovaj operacijski sustav podržava već pomalo zaboravljeni assembler, BASIC, C i C++.

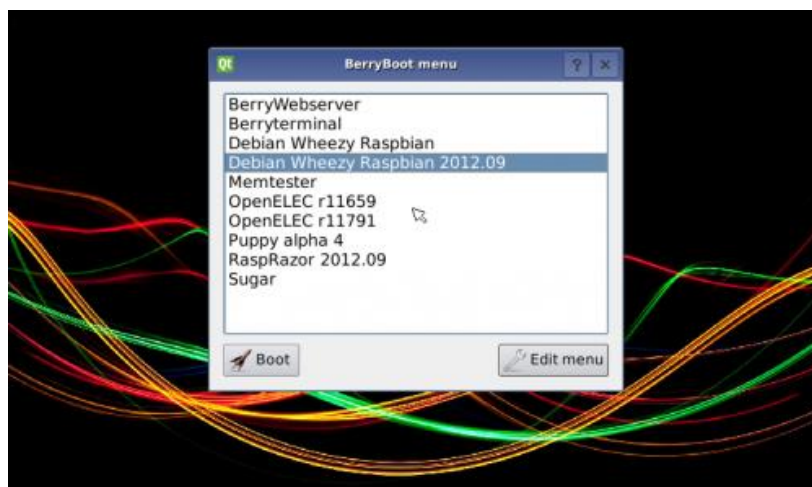
4.4.4. Značajke sustava

Jedina značajka ovog sustava je njegova brzina. Svi korisnički programi, pa tako i sam operacijski sustav iznimno su memorijski ne zahtjevni, stoga se sve radnje odvijaju gotovo bez čekanja. To je rezultat razvoja operacijskog sustava iz doba kada su računala raspolagala sa procesorima radnog takta na 8MHz i 512KB radne memorije. U današnjim okvirima, pa čak i na Raspberry Pi računalu to znači pregršt slobodnih resursa za izvršavanje procesa.

4.5. BerryBoot

Kako bi što bolje iskoristili mogućnosti vlastitog računala ili zbog prirode posla, najčešće u nekim razvojnim poduzećima, korisnici često za radno okruženje koriste više od jednog operacijskog sustava. Mogućnost je to koju je nakon šest mjeseci od izlaska originalnog Raspberry Pi računala na tržištu omogućio operacijski sustav pod imenom Berryboot.

Taj operacijski sustav temeljen na Linux jezgri danas služi više kao pomoćni alat nego kao sam operacijski sustav. Donosi jednostavan i fleksibilan način promjene radne okoline koristeći memorijske kartice, USB memorije ili pak mrežne lokacije.²⁰



Slika 27: Berryboot glavni izbornik

4.5.1. Instalacija operacijskog sustava

Nakon raspakiravanja paketa koji sadrži BerryBoot operacijski sustav na memorijsku karticu, pokretanjem na Raspberry Pi računalu korisnika se vodi kroz niz koraka za postavljanje osnovnih konfiguracijskih parametara kao što su odabir žične ili bežične konekcije, postavke lokalizacije, odabira i formatiranja memorijskih lokacija za instalaciju drugih operacijskih sustava. Nakon završetka instalacije korisnik se preusmjerava na glavni izbornik na kojem se upravlja ostalim operacijskim sustavima.

²⁰ BerryBoot v2.0 – bootloader, 2020. <https://www.berryterminal.com/doku.php/berryboot> (dostupno 21.04.2020.)

4.5.2. Konfiguracija sustava

Osim osnovnih konfiguracijskih parametara navedenih u prethodnom poglavlju korisnik ima mogućnost odabira automatskog ili ručnog odabira željenog operacijskog sustava za pokretanje nakon paljenja računala. Ima mogućnost postavljanja vremena nakon kojega će se odabrani operacijski sustav sam pokrenuti, te kraju ima mogućnost odabira favorita, odnosno slaganja liste operacijskih sustava po prioritetu korištenja. Još jedna od značajki je postavljanje lozinke za onemogućavanje neautoriziranog uređivanja ili brisanja operacijskih sustava sa medija.

4.5.3. Korisnički programi

Obzirom da BerryBoot nije namijenjen za rad, već isključivo kao pomoć pri radu sa ostalim operacijskim sustavima, osim uređivača teksta za izvođenje promjena nad konfiguracijskim parametrima ne posjeduje dodatne korisničke programe.

4.5.4. Značajke sustava

Cijeli sustav bazira se na upravitelju zaduženom za manipulaciju drugim operacijskim sustavima. Omogućava korisniku jednostavno dodavanje novih operacijskih sustava sve dok korisnik ima dovoljno slobodne memorije na odabranom mediju. Instalacija se može izvršiti na memorijsku karticu, eksterni tvrdi disk, eksternu USB memoriju ili na neku mrežnu lokaciju. Na jednostavan način omogućava korisniku stvaranje kopije operacijskog sustava u njegovom izvornom obliku ili zajedno sa promjenama koje je korisnik u njemu napravio. Osim toga omogućava stvaranje rezervi (eng. backup) sustava na drugim prijenosnim medijima kako bi se sačuvali podaci u slučaju da se nešto dogodi sa primarnom memorijskom lokacijom.

5. Programski jezici i alati za razvoj programskih rješenja na Raspberry Pi

Programski jezici nekako su prirodno povezani sa Raspberry Pi platformom. Prisjećanjem na izvornu ideju, da se pristupačnim sklopovljem podiže razina informatičke pismenosti u slabije razvijenim zemljama, Raspberry Pi računalom to je moguće postići na 3 načina: upoznavanjem sa ugrađenim sklopovskim komponentama, upoznavanjem sa sučeljem instaliranog operacijskog sustava i ono najvažnije učenjem programskih jezika i razvojem vlastitih aplikacijskih rješenja. Logičan je dakle bio i izbor operacijskog sustava otvorenog koda koji daje mogućnost pogleda „ispod haube“, odnosno uvida u programski kod, otkrivanja njegovih funkcionalnosti, te dodavanja vlastitih nadogradnji. Proces učenja programiranja odnosno usvajanja novih znanja težak je i dugotrajan postao, stoga mnogi često odustaju već nakon samog početka. Ova vještina zahtjeva shvaćanje logičkih koncepata, algoritama, apstraktnih modela i sl. Treba voditi računa i o konstantnom napretku tehnologije i razvoja programskih jezika. Ovo je definitivno dug i kontinuiran proces bez čvrstog i vidljivog cilja. Jedina donekle čvrsta konstanta je sintaksa samog programskog jezika. Osnove su u pravilu uvijek iste, dok se s novim verzijama jezika mijenjaju nešto kompleksniji koncepti koji u konačnici rezultiraju jednostavnijem obavljanju neke funkcije.

Razvijanje programskih proizvoda moguće je u najjednostavnijem uređivaču teksta, kao što je Notepad na Windows platformi ili Gedit na Linux-u. Treba priznati da iako zvuči jednostavno ovo definitivno nije način i smjer u kojem treba ići, odnosno ovakav pristup nema smisla. Jednostavnijem shvaćanju i učenju programskih jezika pomažu adekvatna integrirana razvojna okruženja, odnosno alati koji korisnicima omogućuju kvalitetnije i brže pisanje koda. Razvojne okoline danas sadrže ugrađene predloške za mnoge programske jezike, pa tako samim provođenjem statičke analize i formatiranja, kod postaje mnogo jednostavniji i razumljiviji početnicima. Osim statičke analize koda nude i ugrađene kompilatore koji na osnovu zadanih pravila svakog podržanog programskog jezika u mogućnosti su prije samog izvođenja ili za vrijeme pokretanja koda uočiti pogreške i predložiti promjene. Kroz neka od razvojnih okruženja, kao i najpopularnije programske jezike podržane na Raspberry Pi platformi proći će se u idućim poglavljima.

5.1. Podržani programski jezici

Raspberry Pi platforma kao što smo već vidjeli svestrana je u smislu podrške operacijskih sustava, samim tim i podržanih programskih jezika. Programski jezici nastali su kao odgovor na potrebu za bržim i efikasnijim pisanjem koda, zamijenivši binarne instrukcije strojnog jezika mnemoničkim oznakama. Danas od jezika niže razine u upotrebi ostao je u najvećoj mjeri asemblerski jezik. Ovaj jezik daje potpunu kontrolu programerima nad sklopovljem računala, ali je strogo vezan za računalnu arhitekturu. To znači da ovakav program nije prenosiv među računalima različite arhitekture. Iz ovog razloga danas se koriste jezici više razine, odnosno jezici koji prevladavaju sve nedostatke asemblerskog jezika. Programer se oslobađa znanja oko arhitekture sustava i usmjeren je samo na pisanje koda koji je u konačnici u potpunosti prenosiv tj. neovisan o arhitekturi. U nastavku poglavlja nalazi se pregled najvažnijih programskih jezika podržanih na službenom operacijskom sustavu Raspbian.



Slika 28: Programski jezici C++, Java, Python

5.1.1. Programski jezik C

C programski jezik viši je proceduralni programski jezik opće namjene. Razvio ga je Dennis Ritchie sedamdesetih godina 20. stoljeća. Popularnost je dosegao vrlo brzo, pa ga je 1990. godine prihvatila i Međunarodna organizacija za standarde. Bio je to prvi programski jezik koji je omogućio brzo učenje, pisanjem naredbi koje se prevode u strojni jezik, a omogućio je i modularno pisanje programa. Većim dijelom svi Linux operacijski sustavi pisani su u C programskom jeziku, stoga je logično da ga u potpunosti podržavaju odmah po instalaciji. Za pisanje programa u C-u potreban nam je običan uređivač teksta, a datoteku koja sadrži program potrebno je spremiti sa ekstenzijom `.c`. Pokretanjem prevodioca (*eng. interpreter*) napisana datoteka pretvara se u izvršni kod. Izvršni kod sprema se u drugu datoteku sa imenom `a.out`, samo ako je prevođenje uspješno provedeno. Ukoliko prevođenje nije bilo

uspješno prevodilac će nam javiti greške. U tom trenutku kreće često postupak pronalaženja i otklanjanja grešaka. Kao i u svakom drugom programskom jeziku ovdje je moguće naići na tri vrste grešaka: sintaktičke, logičke i funkcionalne. U slučaju sintaktičke greške prevodilac će odmah prekinuti izvođenje prevođenja i javiti grešku. Kod logičke greške program se može normalno pokrenuti, može izvršavati svoju funkciju i u nekom slučaju prekinuti svoje izvršavanje. U tom slučaju operacijski sustav će sam najčešće javiti grešku o neispravno prekinutom izvođenju programa. Treći slučaj je kada program u potpunosti radi, ali ne izvršava zamišljenu funkciju programera.

Bilo kako bilo u C programskom jeziku prilikom otklanjanja grešaka korisnik se može poslužiti dvama programima. Jedan od njih naziva se „lint“ koji radi statičku analizu napisanog koda i provjerava sintaksu, dok drugi pod imenom „debugger“ omogućava izvođenje programa sa zadanim prekidima, odnosno izvođenje programa liniju po liniju. To je ujedno i najefikasniji način otkrivanja i uklanjanja grešaka neovisno o kojem programskom jeziku se govori. C programski jezik omogućava da se različiti dijelovi koda pišu u različitim datotekama, koje će se na kraju prilikom prevođenja spojiti u jednu. Ovu značajku omogućuje povezičavač (*eng. linker*) odnosno alat za povezivanje izvornog koda i datoteka zaglavlja. Datoteke zaglavlja vrlo je lako prepoznati po ekstenziji `.h`. One se obično koriste za proširivanje mogućnosti koda i potrebne su prevodiocu za uspješno prevođenje programa.²¹

- Primjer jednostavnog Hello World programa napisan u C programskom jeziku:

```
#include<stdio.h>
int main(void)
{
    printf("Hello World! \n");
    return 0;
}
```

U ovom jednostavnom programu na primjeru moguće je vidjeti uključivanje datoteke zaglavlja sa imenom *stdio.h*, koja sadrži funkcije za upis i ispis podataka na standardni ulaz odnosno izlaz. Definiciju glavne funkcije *int main()* koja definira tip povratne vrijednosti cijelog

²¹ J. Mladen, *Programski jezik C*, 2004.

broja putem oznake *int* i koja ne prima nikakve argumente, označeno sa *void*. Nadalje naredba *printf* je funkcija iz datoteke zaglavlja za ispis na standardni izlaz koja prima argumente u obliku teksta okruženog dvostrukim navodnicima. Program završavamo vraćanjem vrijednosti 0 koja operacijskom sustavu znači uspješno izvršavanje programa, dok se svaka druga vrijednost smatra greškom, odnosno neuspješnim izvođenjem.

Programski jezik C od ostalih programskih konstrukata omogućuje korištenje varijabli, petlji, uvjeta grananja, polja znakova, pokazivača, operatore i izraze, strukture, datoteke i operacije nad bitovima.

```
Dino@Dino-PC MINGW64 ~/Desktop
$ g++ HelloWorld.c

Dino@Dino-PC MINGW64 ~/Desktop
$ ./a.exe
Hello world!
Dino@Dino-PC MINGW64 ~/Desktop
$
```

Slika 29: Kompiliranje i pokretanje programa pisanog u C-u

5.1.2. Programski jezik C++

C++ programski jezik proširenje je C programskog jezika. Jednostavnim matematičkim rječnikom može se reći da je programski jezik C++ nadskup od C. U programiranje uvodi objektnu paradigmu putem nasljeđivanja, polimorfizma, enkapsulacije i skrivanjem podataka. Osmislio ga je Bjarne Stroustrup 1979. godine za vrijeme pisanja doktorske disertacije. U početku nosio je ime „C with classes“ odnosno C sa klasama. Prva komercijalno dostupna verzija C++ programskog jezika svijetlo dana ugledala je 1985. godine. Pojam objekta objedinio je podatke i operacije nad tim podacima, čime je kod učinio još čitljivijim i razumljivijim, a u programiranje uveo pojam ponovne iskoristivosti. Danas se još uvijek koristi za programiranje računalnih igara, mrežnih i poslužiteljskih aplikacija ponajprije zbog svoje brzine. Veliku novost programiranju Stroustrup je donio definiranjem znakovnih i logičkih tipova podataka pod imenima *string* i *boolean*. *String* predstavlja niz nepromjenjivih znakova, dok *boolean* može predstavljati dva stanja odnosno dvije vrijednosti: istina ili laž.

Osim ugrađenih tipova podataka, jezik je omogućio i korisnički definirane tipove podataka. Kod pronalaženja grešaka u kodu omogućio je takozvano hvatanje iznimaka, koje su omogućile kontrolu toka programa bez prekidanja izvođenja. Naime ovaj kontrolirani tok

omogućio je kao i kod uvjetnog grananja po prvi puta nastavak izvođenja programa predviđanjem mogućih grešaka, odnosno prekidanjem izvođenja programa sa detaljnijim dokumentiranjem podataka o grešci ukoliko se ona dogodila u trenutku kritičnom za izvođenje programa. Uveo je i pojam preopterećivanja funkcija (*eng. function overloading*) gdje su funkcije s istim imenom mogle obavljati različite operacije. Takve funkcije razlikovale su se po argumentima koje primaju. Poziv glavne programske funkcije *main()* onemogućen je kroz ostale programski definirane funkcije.

Tako se došlo do striktno hijerarhije kakvu su kasnije implementirali mnogi drugi programski jezici. Za razliku od C programskog jezika deklariranje varijabli omogućeno je u bilo kojem programskom dijelu, a ne samo na početku programa. Sve ove promjene utjecale su i na broj ključnih riječi. Naime od C programskog jezika koji je imao definirane samo 32 ključne riječi, C++ koristi čak 20 više odnosno 52. U većini programskih jezika ključne riječi označavaju programski konstrukt čija je funkcionalnost odnosno upotreba gotovo jednoznačno određena. Razlika će se pronaći u Python programskom jeziku koji programeru daje potpunu kontrolu i omogućuje manipulaciju čak i sa ključnim riječima programskog jezika.²²

Primjer jednostavnog Hello World programa napisan u C++ programskom jeziku:

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World!";
    return 0;
}
```

Kao i kod C programskog jezika prvo se uključuje datoteka zaglavlja koja se koristi za omogućavanje upisa i ispisa na standardni ulaz odnosno izlaz. Kako bi se u kodu izbjeglo korištenje operatora dodjele područja koristi se naredba *using namespace* koja u program uključuje funkcije iz biblioteke *std*. Bez naredbe *using namespace* u kodu bi moralo pisati *std::cout*. Nakon toga slijedi definicija glavne funkcije programa u kojoj se naredbom *cout* prosljeđuje zadani tekst na standardni izlaz. Na kraju se također za označavanje uspješno izvedenog programa vraća vrijednost 0.

²² S. Fajković, *Uvod u programiranje*, 2015.

```
Dino@Dino-PC MINGW64 ~/Desktop
$ g++ helloworld.cpp

Dino@Dino-PC MINGW64 ~/Desktop
$ ./a.exe
Hello world!

Dino@Dino-PC MINGW64 ~/Desktop
$ |
```

Slika 30: Kompiliranje i pokretanje programa u C++

5.1.3. Python

Python je interpretirani, objektno orijentirani programski jezik opće namijene. Ime je dobio po legendarnoj humorističnoj seriji *Monty Python*, a izmislio ga je Guido van Rossum 1989. godine. Prva komercijalno dostupna verzija sa oznakom 0.9.0 svijetlo dana ugledala je početkom devedesetih godina. Podržavala je napredne koncepte nasljeđivanja klasa, hvatanja iznimaka, složenih tipova podataka i modula. Zbog svoje jednostavnosti i snage danas se koristi u svim sferama programskih rješenja, dok dominaciju ostvaruje na područjima matematičkih i znanstvenih aplikacija, računalnih igara i interneta stvari. Dolazi pred instaliran na većini Linux i Mac operacijskih sustava. Uz instalaciju programskog jezika dolazi i vlastito razvojno okruženje IDLE u koje su uključeni prevodilac, uređivač teksta i alati za testiranje.

Od klasičnih programskih jezika razlikuje se u načinu izvršavanja. Kod interpreterskih jezika kod se izvršava direktno bez potrebe za kompiliranjem prije izvođenja. Python se od ostalih programskih jezika ističe dodatnim programskim strukturama i specifičnostima kao što su ispis liste u obrnutom smjeru, zamjena dviju varijabli u jednom retku, ulančavanje operatora usporedbe, ternarni operator, vraćanje više vrijednosti iz funkcije, itd. Neke od tih funkcionalnosti kasnije su implementirali ostali programski jezici kako bi ubrzali obavljanje određenih funkcija. Od naprednijih struktura omogućava pisanje bezimenih odnosno lambda funkcija, koja se u Python programskom jeziku tumače kao objekti. Označavaju se ključnom riječi *lambda* i omogućuju korištenje neograničenog broja argumenata.

Primjer implementacije koda na klasičan način:

```
def pomnoziSaDva(x):
    return x*2
print(pomnoziSaDva(5))
```

Primjer implementacije korištenjem bezimene funkcije:

```
pomnoziSaDva = lambda x: x*2
print(pomnoziSaDva(5))
```

Ono što Python čini jednostavnijim za učenje od drugih programskih jezika jest činjenica da sam programski jezik ne zahtijeva eksplicitnu definiciju tipa varijable, liste i sl. Korisnik je u mogućnosti mijenjati tip varijable ovisno o upisanom podatku. Tako se može dogoditi da varijablu koja se na početku koristi za spremanje cijelog broja, sada koristi za spremanje cijele rečenice. Osim eksplicitnog definiranja tipa varijabli, korisnik je u mogućnosti „pregaziti“ gotovo sve predefinirane ključne riječi. Ovo kasnije u programiranju može predstavljati veliki problem ukoliko se ne korisnik ne drži konvencije za pisanje koda, koja je detaljno objašnjena na službenim stranicama. Još jedna neobičnost vezana je za korištenje zagrada za odvajanje dijelova koda odnosno definiranje funkcija i upravljanja tokom programa. Naime Python zagrade koristi isključivo za definiranje n-torki, lista i skupova dok za definiranje funkcija i upravljanjem tokom programa koristi isključivo razmake.²³

- Primjer jednostavnog Hello World programa napisan u Python programskom jeziku:

```
print("Hello World!")
```

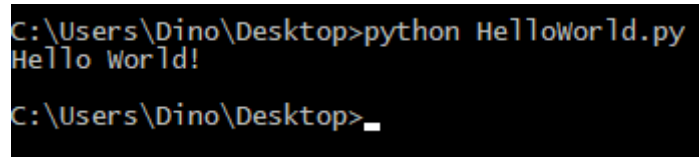
Što reći, jednostavnost prije svega. Osim ključne riječi *print* ne koristi se definiranje glavne funkcije, uključivanje datoteka zaglavlja i sl. Za sve ostale aktivnosti brine se sam prevodilac. Naravno to samo vrijedi kod ovakvih jednostavnih slučajeva. Ispravnije bi bilo definirati glavnu funkciju sa svim popratnim elementima:

```
def main():
    print("Hello World!")

if __name__ == "__main__":
    main()
```

²³ R. Blum and C. Bresnahan, *Sams teach yourself Python programming for Raspberry Pi in 24 hours*. Indianapolis, Indiana: Sams, 2014.

Ovako napisan program moguće je spremiti u datoteku sa ekstenzijom .py i pozvati sa komandne linije pozivom prevodiocu sa ključnom riječi *python* i imenom datoteke kao argumentom.



```
C:\Users\Dino\Desktop>python HelloWorld.py
Hello world!
C:\Users\Dino\Desktop>_
```

Slika 31: Pokretanja programa pisanog u Python-u

5.1.4. Java

Java je danas najkorišteniji objektno orijentirani programski jezik visokih performansi, potpuno neovisan o arhitekturi. Pisan za izvođenje u tzv. Java virtualnom stroju. Inicijalno razvijen za korištenje u multimedijskoj industriji pod nazivom Oak. Direktnan je potomak C++ odnosno C programskih jezika. Početkom devedesetih godina temelje za novi programski jezik postavili su James Gosling, Patrick Naughton, Chris Warth, Ed Frank i Mike Sheridan. Prva komercijalno dostupna verzija pojavila se 1995. godine, a bila je rezultat rješavanja potrebe za jezikom koji će biti u potpunosti prenosiv, kako bi se mogao koristiti na raznim elektroničkim uređajima. Ne izvršava se direktno na računalu kao ostali programski jezici već na standardiziranoj platformi koju nazivamo java okruženje za izvođenje (*eng. java runtime environment*) koje programski implementira virtualno računalo.

Okruženje se sastoji od 2 dijela: java virtualnog stroja i java aplikacijskog programskog sučelja. Java virtualni stroj zadužen je za izvođenje koda, dok java aplikacijsko programsko sučelje implementira klase koje proširuju funkcionalnosti jezika. Prevedeni kod od strane Java prevodioca nazivamo *bytecode*. Bytecode je u potpunosti prijenosan i može ga se pokrenuti na bilo kojem računalu koje posjeduje Java virtualni stroj. Bytecode se može shvatiti kao ekvivalent izvršnom kodu kod drugih programskih jezika. Prevedeni java program spremljen je na disku u više datoteka. Točnije svaka klasa smještena je u svoju datoteku sa ekstenzijom *.class*. Datoteke su obično spremljene u direktorije. Svaki direktorij označava jedan paket. Uključivanje dodatnih klasa u program izvodi se navođenjem putanje do paketa koji sadrži odabranu klasu. Uključivanje se obično izvodi na razini jedne klase ili na razini cijelog paketa. Takva sistematizacija koda omogućuje prevođenje samo onih datoteka na kojima je korisnik radio izmjene, za razliku od programskog jezika C++ kod kojega se ponovnim pokretanjem prevode sve klase. Dinamičnost jezika postiže se učitavanjem potrebnih klasa sa diska ili

mrežne lokacije kada se za to javi potreba. Ova značajka omogućila je razvoj distribuiranih sustava i pozive udaljenih metoda.²⁴

- Primjer jednostavnog Hello World programa napisan u Java programskom jeziku:

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

Program je spremljen u datoteku sa imenom klase i nastavkom `.java`. Prevođenje se iz komandne linije obavlja pozivom prevoditelju sa `javac` i imenom datoteke kao argumentom. Tada se kreira izvršna datoteka istog imena s nastavkom `.class`. Pokretanje programa iz komandne linije izvršava se upisivanjem naredbe `java` i imenom datoteke kao argumentom, ali ovaj puta bez ekstenzije. Ovaj jednostavni Java program sastoji se od javne klase imena `HelloWorld` koja sadrži glavnu funkciju označenu sa `public static void main(String args[])`. Tijelo klase može sadržavati atribute i funkcije. Glavna funkcija `main` neophodna je za pokretanje programa. U glavnoj funkciji naći će se jedna naredba u kojoj se poziva metoda za ispis na standardni izlaz `System.out.println` koja ispisuje proslijeđene argumente, u ovom slučaju tekst „Hello World!“. Ova stroga sintaksa temelj je mehanizma rane i kasne dinamičke provjere koda za vrijeme izvršavanja. Bez obzira na veličinu programa struktura koda zadana je konvencijom i uvijek ostaje ista. Od sigurnosnih mehanizama implementirano je rješenje koje onemogućava dijeljenje memorijske lokacije na dvije varijable, također ugrađene su metoda provjere indeksa polja i niza znakova. Sigurnosni mehanizmi omogućuje provjeru digitalnog potpisa i na osnovu rezultata provjere zaključuje ima li program prava pristupiti određenim resursima.

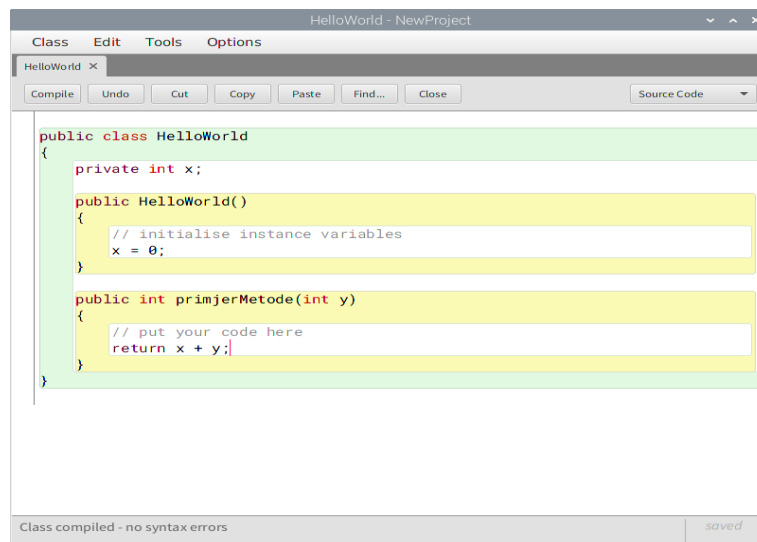
²⁴ M. Čupić, *Programiranje u Javi*, 2015.

```
C:\Users\Dino\Desktop>javac HelloWorld.java
C:\Users\Dino\Desktop>java HelloWorld
Hello world!
C:\Users\Dino\Desktop>_
```

Slika 32: Kompiliranje i pokretanje programa u Javi

5.2. BlueJ

Integrirano je razvojno okruženje namijenjeno razvijanju aplikacijskih rješenja u Javi. Primarno je namijenjen početnicima koji tek ulaze u svijet objektno orijentiranog programiranja. Alat je razvijen na King's College u Londonu i licenciran je kao alat otvorenog koda licencom GNU GPL verzije 2. Alat je dostupan na Windows, Mac i Linux platformama. U slučaju Raspbian operacijskog sustava za Raspberry Pi u proširenom paketu dolazi pred instaliran i spreman za korištenje. Program posjeduje i tzv. samostalnu (*eng. standalone*) verziju koja dolazi sa Java razvojnim okruženjem, neovisna je o platformi i moguće ju je pokretati sa eksternih medija kao što su USB memorije.²⁵



Slika 33: Uređivač teksta na BlueJ okolini

Nakon instalacije i pokretanja korisnik se nalazi u glavom prozoru, odakle može kreirati novi ili otvoriti postojeći projekt. Neovisno o odabranoj akciji nakon otvaranja/kreiranja projekta

²⁵ BlueJ documentation, 2020. <https://www.bluej.org/doc/documentation.html> (dostupno 21.04.2020.)

u glavnom prozoru pojavljuje se prikaz hijerarhije klasa. Ovdje je korisnik u mogućnosti pritiskom na gumb kompilirati postojeće rješenje, kreirati nove klase, sučelja ili pak jedinične testove. Pokretanje programa vrši se također putem grafičkog sučelja, odabirom klase i metode koja se želi pokrenuti. Zanimljivost ovog prozora je mogućnost korisnika da jednostavnim klikom miša označi odnose među klasama, odnosno grafičkim putem kreira nasljeđivanja unutar projekta. Otvaranjem datoteke klase ugrađeni uređivač teksta različitim bojama označava dijelove koda kako bi isti bio što čitljiviji.

Zelenim kvadratom označeno je područje klase, dok je žutim kvadratom označeno područje metode. Uređivač teksta sadrži integrirane akcije formatiranja koda, umetanja komentara, pronalaženja u tekstu i postavljanja prijelomnih točaka koje se koriste tijekom otklanjanja grešaka u kodu. Kako bi se još više naglasio edukacijski aspekt ovog alata, svaki kreirani objekt ima i svoju grafičku reprezentaciju nad kojom korisnik može desnim klikom odabrati metode implementirane u klasi objekta. Zanimljivo je i to što korisnik može sam imenovati instance kreiranih objekata radi jednostavnijeg snalaženja.

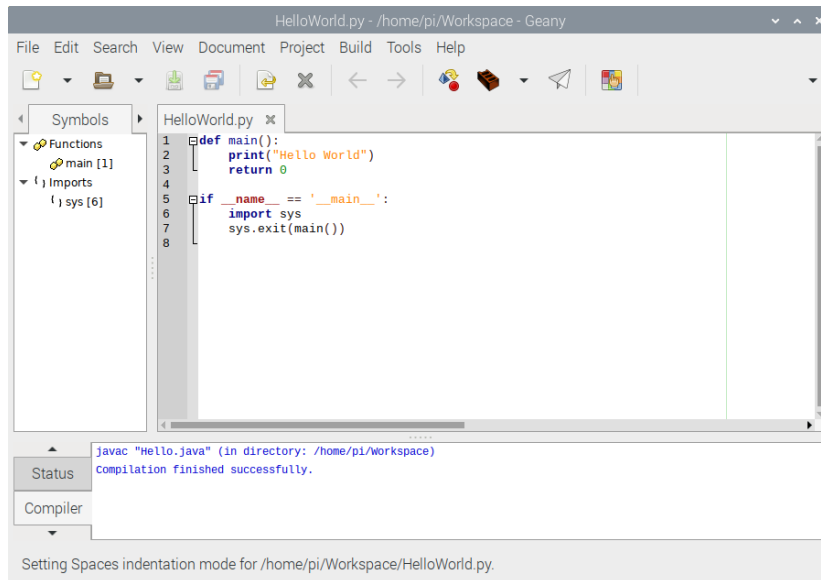
5.3. GeanyIDE

Još jedna u nizu integriranih razvojnih okolina koje dolaze pred instalirane na Raspbian operativnom sustavu. Za razliku od BlueJ razvojne okoline GeanyIDE nije ograničen na jedan programski jezik, već ih po službenoj dokumentaciji podržava čak 68. Alat je dostupan na Linux, FreeBSD, NetBSD, OpenBSD, MacOS X, AIX v5.3, Solaris Express i Windows operacijskim sustavima i u potpunosti je otvorenog koda. Posjeduje i vlastitu knjižnicu sa proširenjima koja je moguće instalirati kako bi omogućili dodatne funkcionalnosti u alatu. Alat je moguće koristiti kroz integrirano grafičko sučelje ili putem komande linije.²⁶

Grafičko sučelje omogućuje korisniku sve funkcionalnosti jednog modernog razvojnog okruženja od kreiranje datoteka iz predložaka, automatskog formatiranja koda ovisno o jeziku, do nebrojenih mogućnosti personalizacije samog korisničkog sučelja. Ukoliko na korisničkom računalu postoji instalirana potrebna korisnička podrška u obliku odgovarajućih kompilatora i prevoditelja, programe je moguće pokretati i iz okruženja. Glavni prozor razvojnog okruženja

²⁶ T. Enrico, T.Nick, *Geany*, 2019. <https://www.geany.org/manual/current/index.html> (dostupno 21.04.2020.)

podijeljen je na 6 glavnih cjelina: glavni izbornik, alatna traka, alatne trake sa prikazom dokumenata u radom direktoriju, područja tekstualnog uređivača, područja za ispis poruka i trake statusa.



Slika 34: Geany razvojno okruženje

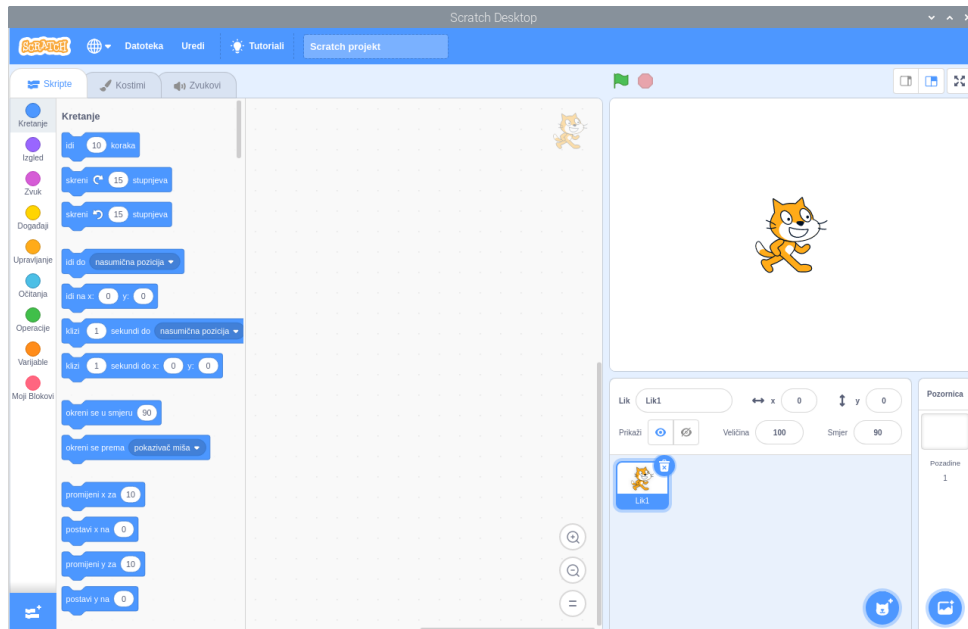
Glavni izbornik sadrži osnovne akcije za upravljanje alatom, alatna traka sadrži prečice za upravljanje izvođenjem programa, kompiliranjem i sl. Tekstualni uređivač u stanju je formatirati kod po predlošku, odnosno po odabranom jeziku pisanja. Dopunska alatna traka kao u modernijim integriranim okruženjima pokazuje glavne programske konstrukte poput funkcija, objekata, korištenih biblioteka i sl.

5.4. Scratch

Scratch je interaktivna razvojna okolina namijenjena najmlađim korisnicima Raspberry Pi računala. Odnosno korisnicima od 8 do 16 godina kako navode službene stranice. Omogućuje kreiranje interaktivnih priča pomoću sustava „povuci i pusti“. Potiče logičko razmišljanje i stvara podlogu za korištenje klasičnih programskih jezika. Jezik kao i sama okolina nastala je na MIT-u (eng. *Massachusetts Institute of Technology*) i potpuno je besplatna. Otvorenog je koda stoga ju je moguće mijenjati i prilagođavati. Jezik je neovisan o platformi i osim putem

instalacijskog paketa moguće ga je pokrenuti iz gotovo svih Internet preglednika. Obzirom da se sve operacije izvode nad objektima može se reći da je objektno orijentirani programski jezik.

27



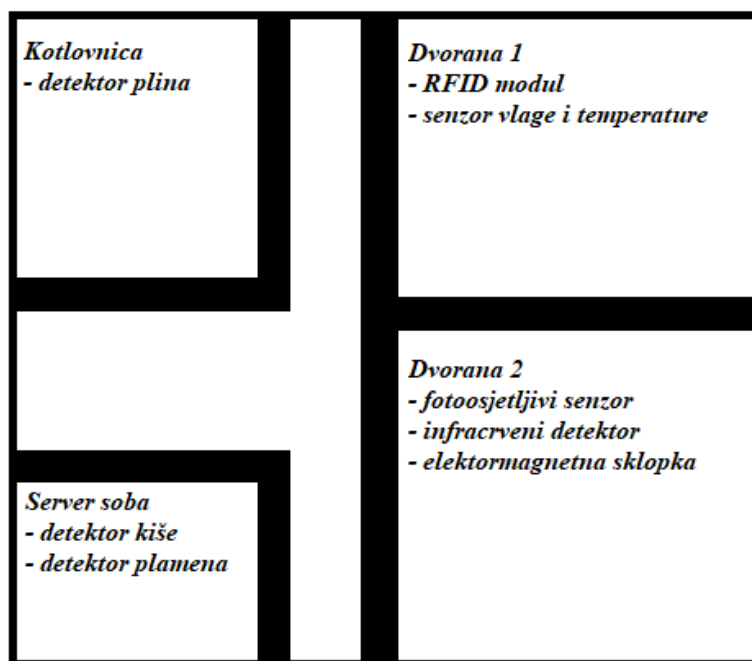
Slika 35: Glavni zaslon Scratch okruženja

Sadrži veliku online zajednicu korisnika koji imaju mogućnost dijeljenja kreiranih aplikacija. Radno okruženje sastoji se od glavnog prozora u kojem korisnici mogu odabrati pred definirane skripte. Skripte upravljaju ponašanjem objekata. Objekti se mogu učítavati iz knjižnice objekata ili ih je moguće kreirati odnosno učítati sa računala. Korisniku je omogućeno u skriptama mijenjati osnovne atribute poput iznosa rotacije i kretanja objekta po zaslonu ili pak naprednijih operacija poput upravljanja efektima zvuka. Raznim elementima koji se dovlače na radnu podlogu programa implementirane su i ostale programske strukture poput uvjeta, petlji, iznimki i sl.

²⁷ Scratch Wiki, 2020. https://en.scratch-wiki.info/wiki/Scratch_Wiki (dostupno 21.04.2020.)

6. Programsko rješenje

Programsko rješenje kojim će se pokazati dio mogućnosti Raspberry Pi računala zamišljeno je kao koncept pametnog sustava koji se može koristiti u obrazovnim, ali i drugim ustanovama u kojima se javlja potreba nadzora studenata, zaposlenika i imovine. Sustav koristi ugrađene mogućnosti ulaza i izlaza opće namjene na Raspberry Pi računalu kako bi putem priključenih senzora ostvarivao interakciju sa korisnicima putem RFID (*eng. Radio-frequency identification*) tehnologije, dok se imovina nadgleda serijom senzora koji su u stanju detektirati razne anomalije u prostoru kao što su vatra, plin, voda i sl. Na Slika 36 prikazan je zamišljeni model prostora sa razmještajem senzora korištenih pri implementaciji ovog pametnog sustava.



Slika 36: Model prostora sa sensorima

6.1. Slučajevi korištenja

Kako bi se što jednostavnije prikazalo korištenje sustava u ovom poglavlju opisani su osnovni slučajevi korištenja koji pobliže opisuju funkcije senzora i njihovu interakciju sa programskim rješenjem na Raspberry Pi računalu, ali i konačno korisničko programsko rješenje vidljivo kroz web aplikaciju na računalu korisnika.

6.1.1. Slučaj korištenja za RFID modul

RFID modul odnosno modul za identifikaciju omogućuje čitanje i pisanje podataka sa elektronskih kartica. Kartice jednoznačno predstavljaju korisnika sustava. U ovom slučaju korisnik je student kojem kartica služi za praćenje prisustvovanja nastavi. Po ulasku u dvoranu student skenira svoju karticu i tako označi prisustvo na predavanju. Ova akcija u bazi podataka ostavlja zapis osnovnih informacija o studentu, oznake za ulaz kao i vremenu ulaska u dvoranu. Na izlazu iz dvorane student također skenira svoju karticu i na taj način u bazu postavlja oznaku izlaza i vrijeme. Na osnovu ta dva vremena sustav izračunava proteklo vrijeme za određeni predmet i dodaje ostalim zapisima o pojedinom studentu. Na korisničkom sučelju web aplikacije kod pregleda informacija o studentu moguće je vidjeti sve predmete koje pohađa odabrani student, broj odslušanih sati po svakom od predmeta i uvjet za odslušanim satima. Po potrebi studentu je moguće poslati kratak sažetak u e-mail poruci o stanju odslušanog fonda sati svih predmeta s ciljem izbjegavanja neostvarivanja uvjeta zbog nedolazaka na predavanje. Sva stanja bilježe se u dnevnik rada aplikacije.

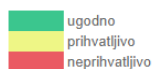
6.1.2. Slučaj korištenja za senzor plina

Senzor plina smješten je u kotlovnici ustanove. Senzor je u mogućnosti detektirati više vrsta plinova kao što su alkoholne pare, propan, metan, vodik, i sl. ali zbog nedostatka dodatnih elektroničkih sklopova koji omogućavaju širi spektar detekcije u ovom sustavu koristi se kao detektor ukapljenog zemnog plina koji se najčešće koristi kao ogrjevno gorivo. U slučaju detekcije curenja obavještava se administrator sustava kroz korisničko sučelje kao i kroz e-mail notifikaciju. Sva stanja bilježe se u dnevnik rada aplikacije.

6.1.3. Slučaj korištenja za senzor vlage i temperature

Senzor temperature i vlage svakih 5 minuta bilježi vrijednosti i zapisuje ih u bazu. Ovisno o vrijednostima prema Tablica 4 pozadinska aplikacija po potrebi podiže alarm i obavještava administratora sustava o potrebnim daljnjim akcijama. Prijedlozi za daljnje postupanje donose se u kombinaciji sa informacijama sa OpenWeatherMap servisa za trenutnu lokaciju. Naime dohvaćena temperatura i vlaga sa senzora uspoređuje se sa stvarnom temperaturom i vlagom za određenu lokaciju i na osnovu danih parametara predlažu se daljnje akcije poput provjetravanja prostorija, paljenja uređaja za ovlaživanje prostora ili pak paljenja rashladnih uređaja. Sva stanja bilježe se u dnevnik rada aplikacije.

Tablica 4: Omjer temperature i vlage u prostoru (preuzeto sa: <https://proluft.hr/savjeti-i-blog/savjeti/kucanstvo/optimalna-temperatura-i-vlaznost-ugodan-boravak-46/>)



Relativna vlažnost	Temperatura				
	26°C	24°C	22°C	20°C	18°C
0%	red	red	red	red	red
10%	red	red	red	red	red
20%	red	yellow	yellow	yellow	red
30%	yellow	yellow	yellow	yellow	red
40%	yellow	yellow	green	green	yellow
50%	red	yellow	green	green	yellow
60%	red	yellow	green	green	green
70%	red	red	yellow	green	green
80%	red	red	red	yellow	yellow
90%	red	red	red	red	red
100%	red	red	red	red	red

6.1.4. Slučaj korištenja za senzor plamena

U zamišljenom prostoru senzor za detekciju plamena postavljen je u prostoriju sa poslužiteljima kako bi na vrijeme detektirao mogući požar i spriječio gubitak digitalne imovine ustanove. Ima jednostavnu funkciju detekcije plamena, a ovisno o rezultatu kroz pozadinsku aplikaciju obavještava administratora sustava putem korisničkog sučelja web aplikacije i putem e-mail poruke. Web aplikacija na korisničkom sučelju ispisuje preporuku o daljnjem postupanju, kao što je poziv vatrogascima i sl. Sva stanja bilježe se u dnevnik rada aplikacije.

6.1.5. Slučaj korištenja za senzor kiše

Senzor kiše odnosno u ovom programskom rješenju senzor za detekciju poplave također se nalazi u sobi sa poslužiteljima i obavlja osnovnu zadaću, a to je na vrijeme detektirati moguće curenje vode i o tome kroz pozadinsku aplikaciju i korisničko sučelje obavijestiti administratora sustava, te predložiti daljnje akcije. Sva stanja također se bilježe u dnevnik rada aplikacije.

6.1.6. Slučaj korištenja za fotoosjetljivi senzor, senzor pokreta i relej

Sinergijom rada ova tri modula u sustavu omogućuju se uštede u potrošnji električne energije. Fotoosjetljivi senzor pri nedostatku svjetlosti u prostoriji obavještava korisnika sustava o dodatnim zahtjevima. Nadalje senzor pokreta zadužen je za praćenje stanja u prostoriji, ukoliko se detektira određena kretanja, a količina svjetlosti u prostoriji nije dovoljna prema informacijama sa fotoosjetljivog senzora putem pozadinske aplikacije daje se signal elektromagnetnog sklopki koja je zadužena za paljenje odnosno gašenje rasvjetnih tijela. Na ovaj način efikasno se upravlja rasvjetnim tijelima kako bi se ostvarila optimalna potrošnja. Vremena paljenja i gašenja rasvjetnih tijela kroz pozadinsku aplikaciju bilježe se u bazu i na temelju njih u korisničkoj aplikaciji moguće je vidjeti okvirni izračun potrošnje električne energije kao i samu cijenu. Sva stanja bilježe se u dnevnik rada aplikacije.

6.2. Senzori

Za izradu ovog pametnog sustava korišteno je ukupno osam različitih elektroničkih sklopova koji su putem ulaza i izlaza opće namijene povezani sa Raspberry Pi računalom. Upravljanje senzorima odnosno dohvaćanje i postavljanje stanja odvija se u pozadinskoj aplikaciji pisanoj u Python programskom jeziku. Aplikacija koristi ugrađene biblioteke Raspberry Pi računala kako bi ostvarila komunikaciju na pojedinim sučeljima. Komunikacija je digitalne prirode i svodi se na praćenje promjene stanja na ulazu odnosno izlazu, a manifestira se kao prisustvo ili odsustvo napona na traženom sučelju.

Kroz ovo poglavlje osim jednostavnih objašnjenja svakog od korištenih senzora proći će se i kroz specifikacije bitne za razvoj jednog ovakvog programskog rješenja, počevši od zahtijeva za napajanjem do točnosti mjerenja i posebnih upozorenja odnosno režima rada svakog pojedinog senzora.

6.2.1. DHT11 – digitalni senzor temperature i vlage

DHT11 digitalni je senzor temperature i vlage. Pripada seriji električnih termootpornika kod kojih se električni otpor mijenja sa promjenom temperature odnosno vlage. Obzirom na vrstu može se svrstati u NTC (engl. *Negative Temperature Coefficient*) termistore koje definira negativni temperaturni koeficijent. Pojednostavljenim rječnikom može se reći da se povećanjem temperature ili vlage električni otpor na senzoru smanjuje. Mjerni raspon senzora temperature kreće se od 0°C do 50°C, a vlage između 20% do 90%. Dozvoljeno odstupanje od izmjerene vrijednosti za temperaturu iznosi 2°C, a za vlagu 5%. Za razvojne potrebe bitno

je naglasiti još jednu specifikaciju, a to je odziv senzora pri mjerenju koji iznosi maksimalnih 15 sekundi kod mjerenja vlage, odnosno maksimalnih 30 sekundi pri mjerenju temperature.



Slika 37: DHT11 digitalni senzor temperature i vlage (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/31EM6oovwLL_SX466.jpg)

Za rad samog senzora potrebno je osigurati stabilan izvor napajanja između 3.0V – 5.5V, što u potpunosti zadovoljavaju sučelja na Raspberry Pi računalu. Nakon spajanja u strujni krug interakciju sa senzorom bitno je odgoditi otprilike 1 sekundu. To stanje smatra se nestabilnim i potrebno je za pravilnu kalibraciju samog senzora, a nepoštivanje ovog pravila može dovesti do neispravnih vrijednosti na izlazu. Za interakciju sa Raspberry Pi računalom koristi tri pina, od kojih su dva zadužena za napajanje senzora, dok se jedan koristi za serijski prijenos podataka prema računalu.²⁸

6.2.2. MQ-2 – senzor plina

MQ2 digitalni je senzor plina koji sadrži kositrov dioksid kao osnovni element koji se koristi za detekciju plinova. Kositrov dioksid na osjetilnoj komponenti u prisustvu plinova mijenja vodljivost, a ta promjena manifestira se dalje kao promjena napona na izlazu. Ta promjena uzima se kao referentna vrijednost za detekciju.

Ova vrsta senzora omogućuje detekciju zemnog ukapljenog plina, metana, propana, vodika i drugih industrijski često korištenih plinova. Senzor za rad zahtijeva stabilan izvor napajanja od 2.5V do 5V, a sa računalom ostvaruje komunikaciju putem dva sučelja, digitalnog i analognog ovisno o implementacijskim potrebama. Od specifičnih zahtjeva preporučuje se

²⁸ Mouser Electronics, "DHT11 Humidity & Temperature Sensor"

po puštanju senzora u rad ostaviti prostora za kalibraciju od otprilike 1 minute kako bi izmjerene vrijednosti bile što točnije.²⁹



Slika 38: MQ-2 digitalni senzor plina (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/51S1R01YqUL. SX342 .jpg>)

6.2.3. YL-83 - Senzor za kišu

Ovaj senzor nudi jednostavan način detekcije kiše odnosno vode. Po principu rada ne razlikuje se previše od senzora temperature i vlage. Senzor putem ugrađenih sklopovskih elemenata detektira razliku u otporu odnosno pad napona na izlazu ovisno o prisustvu odnosno izostanku detektirane vode na osjetilnom elementu senzora. Za rad zahtjeva stabilan izvor napona od 5V, a sa sustavom omogućuje spajanje putem digitalnog ili analognog sučelja. Na samoj komponenti implementiran je potenciometar koji omogućuje fino podešavanje osjetljivosti. Kod korištenja ne zahtijeva kalibraciju već je po puštanju napona odmah spreman za rad.



Slika 39: Senzor za kišu (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/61uO4BlnY-L. SX466 .jpg>)

²⁹ Waveshare, „MQ-2 Gas Sensor User Manual“

6.2.4. KY-026 – Senzor plamena

Senzor plamena jednostavan je optički senzor koji je u stanju detektirati svjetlosne promjene u spektru od 760nm do 1100nm. Iako namijenjen za detekciju plamena bitno je naglasiti da je u idealnim uvjetima senzor potrebno smjestiti na sigurnu udaljenost od izvora plamena kako bi se izbjeglo oštećenje. Naime i ova komponenta kao i mnoge druge osjetljiva je na visoke temperature, koja najčešće uzrokuje kvarove na elektroničkim sklopovima. Najmanja preporučena udaljenost za detekciju iznosi 80 cm. Za rad senzora potrebno je omogućiti stabilan izvor napona od 3.3V do 5V, a širina vidnog polja senzora iznosi 60°. Kao i kod senzora kiše ne zahtjeva se period kalibracije već je detekcija moguća odmah po puštanju u rad.

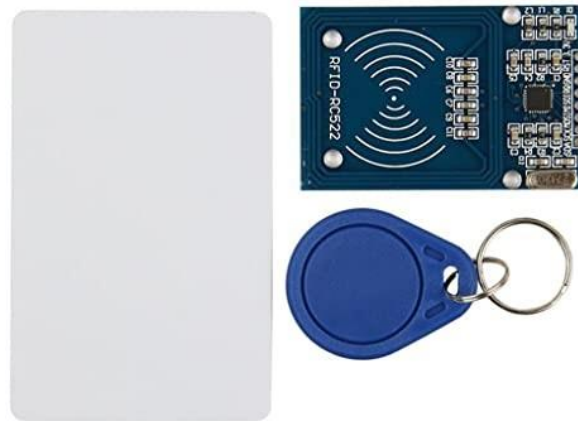


Slika 40: Senzor plamena (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/71wv%2BsRpmWL._SL1500_.jpg)

6.2.5. MF-RC522 – RFID modul

Senzor za beskontaktnu komunikaciju, radne frekvencije od 13,56MHz. Najčešće se koristi kod uređaja koji zahtijevaju komunikaciju uz malu potrošnju električne energije. Implementaciju se najčešće može naći u uređajima za identifikaciju, te pametnim brojidima. Ovaj modul omogućuje dvosmjernu beskontaktnu komunikaciju i prijenos podataka do 424kbit/s. Operativna temperatura za ovaj modul je između 20°C i 80°C, relativna vlaga između 5% i 95%, dok se za napajanje koristi isključivo stabilni istosmjerni napon od 3.3V. U ovom programskom rješenju koristi se za praćenje i identifikaciju korisnika.³⁰

³⁰ "MFRC522 Standard performance MIFARE and NTAG frontend.," 2016.



Slika 41: RFID senzor sa pametnim karticama (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/61H2HVrbU2L.AC.SX466.jpg>)

6.2.6. GL55 – fotoosjetljivi senzor

GL55 odnosno fotoosjetljivi senzor napravljen je od poluvodičkog materijala koji sa promjenom razine svjetlosti mijenja otpor strujnog kruga. Na taj način pad napona na izlazu označava promjenu stanja čija se osjetljivost može podesiti ugrađenim potencijetrom na tiskanoj pločici. Ovaj senzor danas se može pronaći gotovo u svim industrijama počevši od mobilne industrije, industrije igračaka, automobilske industrije, itd. Širok spektar primjena nalazi zbog svojih malih dimenzija, visoke osjetljivosti, brzog odziva i pouzdanosti. Deklarirani radni temperaturni opseg mu je od -30°C do 70°C . U izvedbi za Raspberry Pi računalo napaja se izvorom od 5V. Rezultat mjerenja može isporučiti na dva različita sučelja, digitalno i analogno.³¹



Slika 42: Fotoosjetljivi senzor (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/51OxUXS43FL.SX342.jpg>)

³¹ "GL55 Series Photoresistor Manual"

6.2.7. HC-SR501 – senzor pokreta

Kako mu i samo ime govori omogućuje detekciju pokreta. Koristi se pasivnim elementima odnosno kristalima koji su prirodno polarizirani i pod utjecajem vanjske energije kao što je toplina koju emitira ljudsko tijelo, stvaraju napon. Ova promjena napona indikacija je promjena stanja odnosno detekcije pokreta u prostoru. Ova vrsta senzora najčešće se može pronaći u poslovnim prostorima gdje je ugrađena u alarmni sustav ili u kućanstvima gdje služi kao bežična sklopka za rasvjetna tijela.

Većina ovih senzora omogućuje napajanja od 5V do 12V obzirom pa posjeduju ugrađeni regulator napona koji bez obzira na napon ulaza, izlaz pretvara u napon od 3.3.V. Pogodni su za detekciju kretanja u radijusu od 6 metara, a vidno polje senzora je između 70° i 110°. Široko vidno polje senzora omogućeno je osnovnim fizikalnim svojstvima refrakcije svjetla. Fresnelova leća ugrađena je ispred senzora što omogućuje postavljanje žarišta odnosno fokusa svjetlosti u jednu točku senzora radi veće preciznosti. Osjetljivost senzora detaljnije se može podešavati putem ugrađenog potencijometra. Osim osjetljivosti podešavati se može i vremenski odmak detekcije.³²



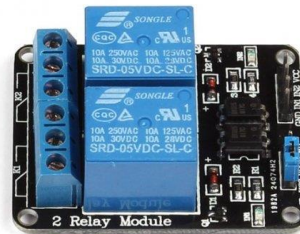
Slika 43: Senzor pokreta (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/61KSJvt3JnL.AC.SX466.jpg>)

6.2.8. Dvokanalna elektromagnetna sklopka

Elektromagnetna sklopka elektromehanička je komponenta upravljana promjenom napona. Može se koristiti kod instalacija izmjenične struje do 250V i 10A, odnosno 30V i 10A kod istosmjerne struje. Na dvokanalnoj elektromagnetnoj sklopki naći će se dva odvojena i individualno upravljana releja koji se mogu koristiti u stanjima kao normalno otvoreni ili

³² A. Lady, "PIR Motion Sensor", 2020.

normalno zatvoreni. Promjenom stanja na ulazu komponente mijenjaju se stanja releja na odabranoj poziciji. Ovaj elektronički sklop zahtijeva izvor vanjskog napajanja od 5V, dok su mu za promjenu stanja na ulazima dovoljni napon od 3.3V i jakost struje od 20mA, iz čega se može vidjeti da je također komponenta koja je energetska učinkovita i savršena za korištenje sa jednostavnim programabilnim mikroupravljačima koji se mogu pronaći na Arduino razvojnim okruženjima odnosno Raspberry Pi računalima.³³



Slika 44: Dvokanalna elektromagnetna sklopka (Preuzeto sa: <https://images-na.ssl-images-amazon.com/images/I/51eyw8a9hKL.jpg>)

6.3. Instalacija i konfiguracija alata za razvoj programskog rješenja

Kroz idućih par stranica proći će se kroz instalaciju okruženja koja su korištenja pri izradi programskog proizvoda pod imenom „SmartSense“ koje je razvijeno kako bi na jednostavan način prikazalo sinergiju korištenja sklopovskih komponenti i programske podrške na Raspberry Pi računalu. Od klasičnih tehnologija i programskih paketa korištene su MySQL i Redis baze podataka, Java platforme i NetBeans razvojnog okruženja te Glassfish poslužitelja za izradu web aplikacije odnosno korisničkog sučelja, dok se od relativno novijih tehnologija koristila Docker platforma za virtualizaciju i Grafana za jednostavnu analitiku i vizualizaciju podataka. Ostali programski paketi kao Raspbian operacijski sustav ili Geany razvojno okruženje za Python objašnjeni su u prethodnim poglavljima.

³³ Handson Technology, “2 Channel 5V Optical Isolated Relay Module“

6.3.1. Docker

Docker platforma poprilično je složen programski paket koji na jednostavan način omogućuje virtualizaciju, odnosno podjelu jednog fizičkog računala na više virtualnih od kojeg svatko ima svoj podskup fizičkih resursa. Kada se govori o virtualnim računalima u Docker kontekstu govori se o kontejnerima sa vlastitim datotečnim sustavom, mrežnim sučeljima i ostalim resursima, ali najčešće bez operacijskog sustava u pozadini. Kontejneri se za izvršavanje funkcija operacijskog sustava oslanjaju direktno na systemske pozive operacijskog sustava domaćina. Na ovaj način jednostavno se postiže efikasnije trošenje sistemskih resursa i brža interpretacija programskih paketa unutar kontejnera.³⁴

Kao osnovni građevni element Docker kontejnera uzimaju se „slike“ koje predstavljaju zapakirane programske pakete spremne za korištenje unutar Docker okruženja. Korisniku je omogućena izrada slika prema vlastitim potrebama što u konačnici rezultira jednostavnijim i bržim postavljanjem kompletnog razvojnog ili izvršnog okruženja. Osim korisnički definiranih slika, Docker posjeduje i službeni registar sa provjerenim i sigurnim slikama pripremljenim za korištenje po principu „out-of-the-box“. Kontejneri u sustavu međusobno su povezani mrežnim podsustavom koji imaju izlaz na Internet, a omogućuju korisniku definiranje mrežnih postavki po vlastitim željama kako bi se efikasno ostvarila među-aplikacijska komunikacija. U izradi programskog rješenja korištene su tri slike sa MySQL, Redis i Grafana programskim paketima. Instalacija Docker okruženja na Raspberry Pi računalo izvodi se preko komandne linije upisivanjem sljedećeg niza komandi:

- Dohvaćanje instalacijske skripte

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

- Pokretanje instalacijske skripte

```
sudo sh get-docker.sh
```

- Dodavanjem ovlasti korisniku za pokretanje kontejnera

```
sudo usermod -aG docker [ime korisnika]
```

Nakon izvršenih komandi Docker sustav trebao bi biti u potpunosti instaliran i spreman za rad, a kako bi se uvjerali da je to sigurno tako mogu se upisati `docker info` ili `docker version`

³⁴ D. Vrčić, *Uvod u Docker*, 2016.

komanda kako bi se osigurale dodatne provjere. Nakon uspješne instalacije idući koraci su ništa drugo nego preuzeti pripremljene slike za Raspberry Pi platformu od programskih paketa koje korisnik želi koristiti. U slučaju ovog programskog proizvoda preuzete su slike od MySQL i Redis baze podataka i Grafana paketa za analitiku. Preuzimanje slika obavlja se naredbom `docker pull` nakon koje slijedi ime slike koja se želi preuzeti, stoga naredbe koje je potrebno unijeti su:

- Dohvaćanje slike MySQL baze podataka

```
docker pull hypriot/rpi-mysql
```

- Dohvaćanje slike Redis baze podataka

```
docker pull redis
```

- Dohvaćanje Grafana programskog paketa

```
docker pull grafana/grafana
```

Provjera skinutih slika u Docker okruženju obavljaju se `docker images` naredbom. Naredba bi trebala vratiti stanje kao na Slika 45. Detaljne informacije o svakoj slici mogu se pogledati naredbom `docker image inspect [ime slike | id slike]`.

```
pi@raspberrypi:~ $ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
redis                latest      632cd87ac6e4     3 weeks ago     72.2MB
grafana/grafana     latest      f9f8382b5821     3 weeks ago     116MB
hypriot/rpi-mysql   latest      4f3cbdbc3bdb     23 months ago   209MB
```

Slika 45: Naredba `docker images`

Nakon što su slike skinute potrebno ih je pokrenuti naredbom `docker run` praćenom sa korisnički definiranim parametrima. Skripta za pokretanje gore spomenutih slika sastoji se od tri `docker run` naredbe, te parametara `-name` za postavljanje razumljivog imena pokrenute slike, `-p` odnosno porta na koji se želi proslijediti komunikacija sa kreiranim Docker kontejnerom, `-v` oznakom koja definira lokalno skladište podataka za odabranu sliku, zastavice `-restart-always` koja Docker sustavu kaže da pokrene slike svakim podizanjem sustava i `-d` oznake koja predstavlja ime slike sa službenog repozitorija na koju se odabrana konfiguracija odnosi.

```
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=admin -v mysql-volume:/var/lib/mysql --restart always -d hypriot/rpi-mysql
```

```
docker run --name redis -p 6379:6379 -v redis-volume:/var/lib/redis --restart
always -d redis
```

```
docker run --name grafana -p 3000:3000 -v grafana-volume:/var/lib/grafana -
-restart always -d grafana/grafana
```

Ako su sve naredbe ispravno upisane slike bi trebale biti pokrenute, a naredba `docker ps` trebala bi dati ispis približan Slika 46 i pokazati sve pokrenute slike na Docker sustavu.

```
pi@raspberrypi:~/Diplomski/Docker_files $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
53af7ead1059   grafana/grafana  "/run.sh"               3 days ago    Up 3 days    0.0.0.0:3000->3000/tcp              grafana
38852b0999dc   redis          "docker-entrypoint.s..." 2 weeks ago   Up 3 days    0.0.0.0:6379->6379/tcp              redis
87391581737f   hypriot/rpi-mysql  "/entrypoint.sh mysql..." 2 weeks ago   Up 3 days    0.0.0.0:3306->3306/tcp              mysql
```

Slika 46: Naredba `docker ps`

Od sada pa nadalje sve baze podataka i Grafana programski paket dostupni su za korištenje na lokalnoj IP adresi i portu definiranom u instalacijskoj skripti ako se njima pristupa sa Raspberry Pi računala odnosno putem eksterne IP adrese svakog od kontejnera zasebno ako se njima pristupa iz neke druge mreže. Ova razina razumijevanja Docker sustava biti će dovoljna za implementaciju programskog rješenja, stoga će se stati sa daljnjim uputama, a čitateljima koji žele dublje ući u tematiku predlaže se proučiti službenu dokumentaciju.

6.3.2. MySQL Workbench

MySQL Workbench alat je za grafičku manipulaciju MySQL poslužiteljima i bazama podataka. U potpunosti je besplatan za sve računalne platforme, a instalacija programskog paketa za Windows platformu u potpunosti je automatizirani proces u kojem se od korisnika traži potvrda lokacije za spremanje svih potrebnih datoteka za izvršavanje programa.

Sam programski paket podijeljen je na 5 cjelina:

- SQL razvoj
- Modeliranje podataka
- Administracija poslužitelja
- Migracija podataka
- Podrška za MySQL Enterprise

Cjelina SQL razvoj omogućava kreiranje i održavanje konekcija sa bazom podataka, definiranje korisničkih parametara i izvršavanje upita nad odabranom bazom. Uz cjelinu za modeliranje podataka najčešće je korišten modul ovog programskog paketa. Druga cjelina

modeliranje podataka na grafički jednostavnom sučelju omogućuje korisniku pregled postojećih baza, kreiranje novih, uređivanje, dodavanje i pregled tablica, atributa i ograničenja nad tablicama. Treća cjelina korisniku omogućuje administraciju korisnika, dodjeljivanje prava i praćenje performansi poslužitelja.

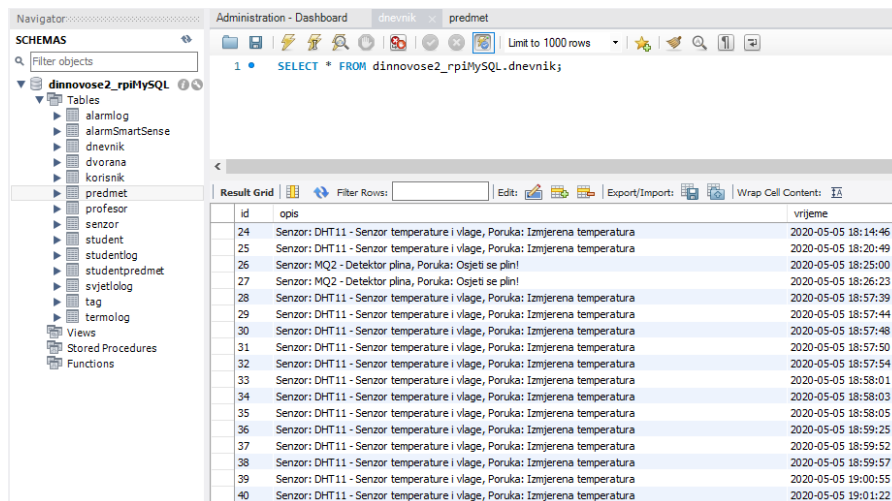
Cjelina migracije podataka ima zadaću ostvarivanja kompatibilnosti za starijim verzijama MySQL poslužitelja odnosno migraciju podataka sa nekih drugih sustava za upravljanje bazama podatka kao što su Microsoft Access, SQLite, PostgreSQL i sl. Zadnja cjelina odnosi se samo na plaćene komercijalne MySQL Workbench programske pakete koji osim standardne korisničke podrške u realnom vremenu imaju dodatne mogućnosti poput definiranja vatrozida i izrade kompletnih sigurnosnih kopija.³⁵

Čitatelju zanimljive funkcionalnosti su kreiranje konekcije prema postojećoj bazi i manipulacija tablicama u grafičkom sučelju. Obzirom da se MySQL baza nalazi na Raspberry Pi računalu, a MySQL Workbench je instaliran na Windows korisničkom računalu, moramo saznati IP adresu Raspberry Pi računala kako bi se uspješno kreirala konekcija na portu definiranom kod pokretanja kontejnera sa slikom MySQL baze. Podaci korišteni za kreiranje konekcije su sljedeći:

```
Connection name: rpiMySQL
Hostname: 192.168.0.15
Port: 3306
Username: dinnovose2
Password: mysqladmin
Default schema: dinnovose2_rpiMySQL
```

Nakon otvaranja nove konekcije prema bazi korisnik se preusmjerava na korisničko sučelje prikazano na Slika 47 odakle ima mogućnosti kreirati nove ili uređivati postojeće tablice i zapise iz baze.

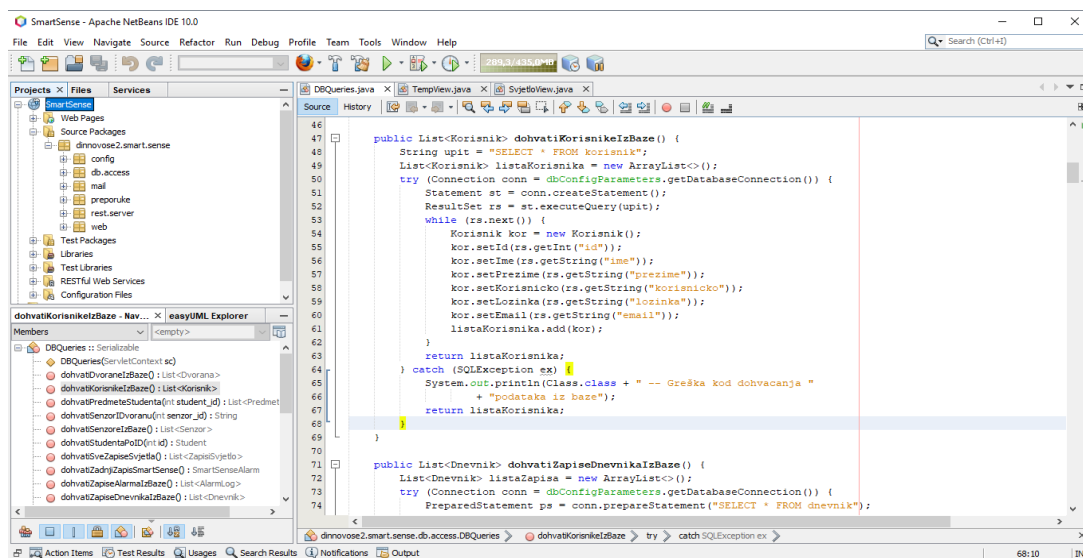
³⁵ "MySQL Workbench reference manual", 2020.



Slika 47: MySQL Workbench korisničko sučelje

6.3.3. NetBeans

NetBeans integrirano je razvojno okruženje namijenjeno izradi i podršci programskih paketa pisanih u Java i JavaEE tehnologiji. Osim podrške za Java tehnologije sadrži i mnoga proširenja koja je moguće dodati prilikom instalacije, a koja omogućuju pisanje koda u drugim programskim jezicima. Alat je u potpunosti besplatan i dostupan na gotovo svim platformama, a projekti napisani unutar okruženja u potpunosti su prenosivi neovisno o platformi. Instalacija programskog paketa na Windows platformi jednostavan je proces u kojemu korisnik odabire željena proširenja, lokaciju instalacije i zadanu verziju Java okoline koja će služiti kao ishodište za pokretanje programa. Za potrebe ovog projekta alat je korišten za administraciju Glassfish poslužitelja te izradu web aplikacije sa korisničkim sučeljem.

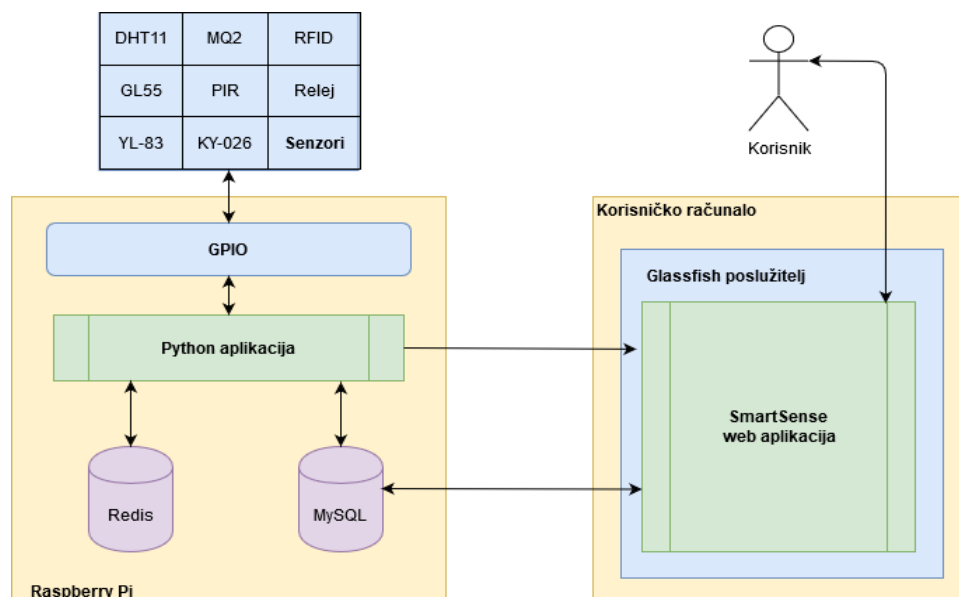


Slika 48: NetBeans razvojno okruženje

Glavni prozor razvojnog okruženja podijeljen je u 4 segmenta. Alatna traka sadrži standardne akcije kao što su kreiranje nove datoteke ili novog projekta, kompiliranje i izvršavanje napisanog programa, definiranje novih konfiguracija ili pokretanje programa u modu za otklanjanje grešaka. Sljedeći segment je prozor sa prikazom trenutnog projekta i stabla direktorija unutar istoga. Omogućuje jednostavnu navigaciju korisnika. Odmah ispod stabla direktorija nalazi se prozor sa objektima trenutno otvorene klase sa glavnog programa. Ovaj prozor sadrži sve konstrukte klase kao što su definirani objekti i metode unutar klase. Nadalje glavni prozor prikazuje trenutnu datoteku na kojoj korisnik radi promjene. Tu se najčešće se radi o klasama nekog paketa ili konfiguracijskim parametrima aplikacije.

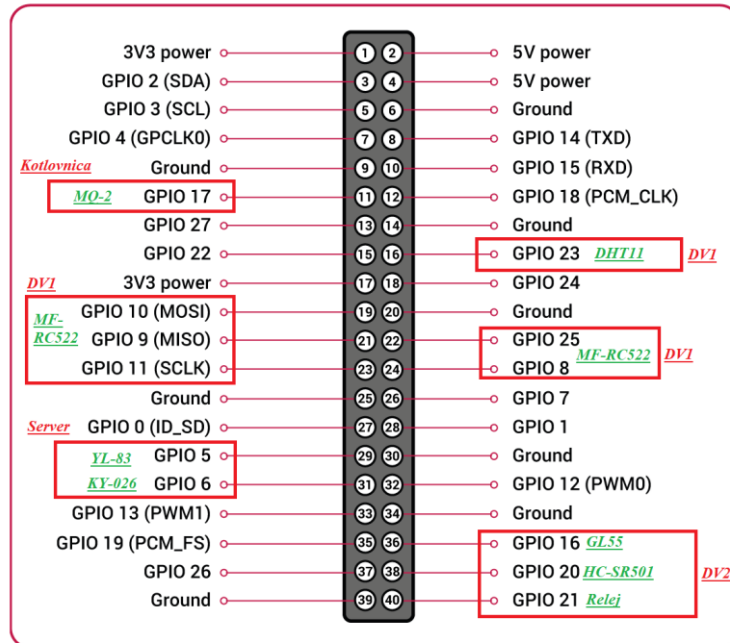
6.4. Arhitektura sustava

Kompletno sklopovsko i aplikacijsko rješenje može se podijeliti na 3 dijela. Prvi dio predstavljaju sve sklopovske komponente koje čine Raspberry Pi računalo i svi priključeni senzori. Zatim slijedi programska potpora koja ostvaruje komunikaciju sa sensorima, a pisana je u Python programskom jeziku i pokreće se direktno sa Raspberry Pi računala. Programsko rješenje komunicira sa Redis i MySQL bazama podataka koje se nalaze unutar Docker kontejnera, a također su pokrenute na Raspberry Pi računalu. Treća komponenta je korisničko računalo odnosno web aplikacija koja komunicira sa Python aplikacijom putem REST servisa, a zadužena je također i za dohvaćanje podataka iz MySQL podataka. Pojednostavljeni model visoke razine apstrakcije cijelog sustava može se vidjeti na Slika 49.



Slika 49: Arhitektura sustava

Povezivanje Raspberry Pi računala i senzora obavlja se preko ulaza/izlaza opće namjene. Kako bi se kasnije jednostavnije pratilo kod samog programskog rješenja na Slika 50 može se vidjeti shematski prikaz GPIO sučelja sa rasporedom senzora prema prostorijama u kojima se nalaze.



Slika 50: Raspored senzora na GPIO sučelju

6.4.1. Redis

Ova baza podataka koja funkcionira po principu ključ – vrijednost u aplikacijskom rješenju služi kao privremeno skladište za podatke i akcije koje u sustavu izazivaju promjene stanja kao što su ulaz i izlaz detektirani na senzoru za identifikaciju, odnosno kao sigurnosna kopija za zapise određenih senzora. Za upravljanje podacima u bazi najčešće se koriste tri ključne riječi: GET, SET i DEL. Naredba `GET [ime_ključa]` dohvaća spremljenu vrijednost za zadani ključ, naredba `DEL [ime_ključa]` briše odabranu vrijednost iz baze dok naredba `SET [ključ vrijednost]` unosi novi zapis u bazu. Ključevi korišteni za spremanje stanja senzora su:

- Sensor_plin
- Sensor_voda
- Sensor_vatra
- Sensor_foto
- Sensor_pokreta
- Serijski broj tag-a

Struktura zapisa za senzore strukturirana je na dva načina ovisno o vrsti senzora kojem zapis pripada, a formatirana je u JSON formatu po principu:

```
{
  "senzor_id: \"id senzora\"": {},
  "poruka: \"Poruka sustava\"": {},
  "vrijeme: \"Vrijeme kada se dogodila promjena stanja\"": {}
}
```

ili

```
{
  "akcija: \"UPALJENO/UGASENO || ULAZ/IZLAZ\"": {},
  "senzor_id: \"id senzora\"": {},
  "poruka: \"Poruka sustava\"": {},
  "vrijeme: \"Vrijeme kada se dogodila promjena stanja\"": {}
}
```

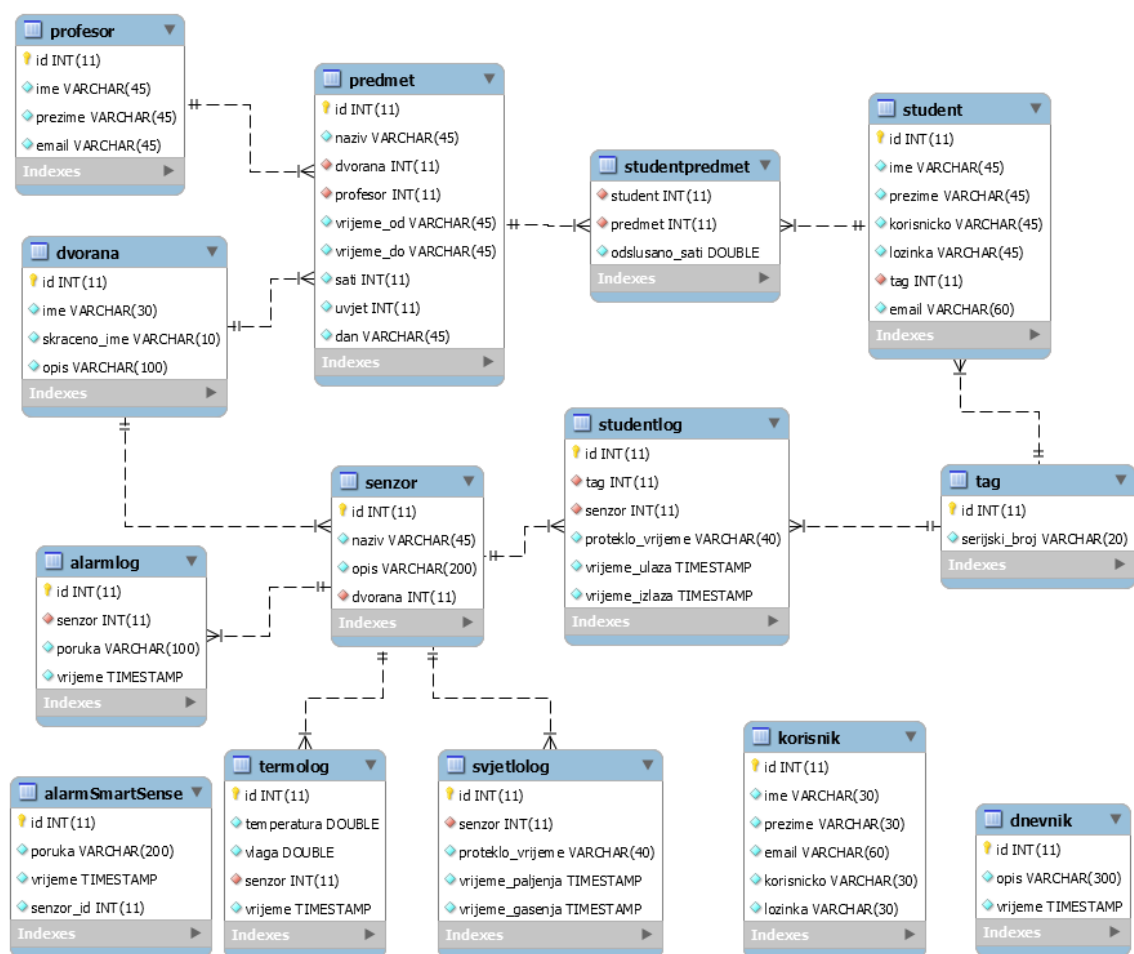
```
127.0.0.1:6379> keys *
1) "Senzor_pokreta"
2) "Senzor_foto"
3) "248673711473"
4) "Senzor_plin"
127.0.0.1:6379> clear
127.0.0.1:6379> keys *
1) "Senzor_pokreta"
2) "Senzor_foto"
3) "248673711473"
4) "Senzor_plin"
127.0.0.1:6379> get 248673711473
"{\"akcija\": \"IZLAZ\", \"tag_id\": 248673711473, \"vrijeme\": \"2020-05-21 13:40:50\"}"
127.0.0.1:6379> get Senzor_pokreta
"{\"akcija\": \"UPALJENO\", \"senzor_id\": 4, \"poruka\": \"Kretanje u dvorani detektirano\", \"vrijeme\": \"2020-05-21 13:40:40\"}"
127.0.0.1:6379> get Senzor_foto
"{\"senzor_id\": 3, \"poruka\": \"Upaliti svjetlo molim\", \"vrijeme\": \"2020-05-21 13:40:26\"}"
127.0.0.1:6379> get Senzor_plin
"{\"senzor_id\": 6, \"poruka\": \"Osjeti se plin!\", \"vrijeme\": \"2020-05-21 13:36:54\"}"
127.0.0.1:6379>
```

Slika 51: Redis baza - primjer dohvaćanja podataka

Kroz Python aplikacijsko rješenje i stanja iz Redis baze o ulazu i izlazu korisnika iz sustava, odnosno paljenja i gašenja rasvjete, na jednostavan način računanjem razlike u vremenu dobivamo zapis koji se upisuje u MySQL bazu i koji kasnije na korisničkom sučelju aplikacije služi za praćenje korisnika sustava i njegovog vremena provedenog u prostorijama ustanove. Dok se u slučaju rasvjete razlika u vremenu koristi za izračun okvirne potrošene električne energije.

6.4.2. MySQL

Dobro nam poznat sustav za upravljanje relacijskim bazama podataka MySQL u ovom projektu koristi se kao osnovna skladišna jedinica za sve podatke koje konzumira web aplikacija kroz korisnička sučelja. Relacijska shema koja pod nazivom dinnovose2_rpiMySQL koja se koristi kao podloga ovog informacijskog sustava sastoji se od ukupno 14 tablica čiji dijagram entiteta i veza se može vidjeti na Slika 52.



Slika 52: Dijagram entiteta i veza MySQL baze

Tablica 5: Popis tablica MySQL baze

Tablica	Opis
profesor	Popis profesora koji predaju u ustanovi
predmet	Popis predmeta koji se predaju u ustanovi
studentpredmet	Označava koji student sluša koji predmet sa odslušanim satima
student	Svi studenti koji pohađaju ustanovu
dvorana	Popis dvorana ustanove
senzor	Popis svih senzora iz sustava sa kratkim opisom i oznakom dvorane
studentlog	Popis vremena boravka studenta po prostorijama
tag	Popis pametnih kartica registriranih u sustavu
alarmSmartSense	Popis alarma pristiglih u aplikaciju REST servisom
termolog	Popis temperature i vlage sa određenog senzora
svjetlog	Popis akcija paljenja i gašenja rasvjete po prostorijama
alarmlog	Popis svih alarma pristiglih u sustav
korisnik	Popis svih korisnika za prijavu u web aplikaciju
dnevnik	Popis svih akcija unutar sustava

6.4.3. Programsko rješenje – Python

Programsko rješenje koje čini osnovu ovog sustava, a izvodi se na Raspberry Pi računalu sastoji se od ukupno 6 modula:

- main.py
- tempCollector.py
- restClient.py
- jsonBuilder.py
- redisConn.py
- mysqlConn.py

Prva dva modula čine osnovu programskog rješenja dok su ostali izvedeni kao pomoćne biblioteke sa specifičnim funkcionalnostima. Zbog poteškoća u sinkronizaciji dretvi u izvođenju glavni program podijeljen je na `main.py` dio koji obavlja sav posao osim dohvaćanja temperature i vlage sa DHT11 senzora. Za taj dio odgovoran je `tempCollector.py` dio koda

koji se oslanja na korištenje vanjske biblioteke „adafruit“ od istoimene kompanije specijalizirane za prodaju elektroničkih komponenti i razvoj programskih proizvoda otvorenog koda. Kako bi se jednostavnije prikazalo kako Raspberry Pi komunicira sa priključenim sensorima proći će se kroz isječke koda koji objašnjavaju tok izvođenja operacija. Za svaki od senzora na početku je kroz poziv funkcije `GPIO.setup()` za određeni pin na GPIO sučelju postavljena njegova zadaća. To znači da će Raspberry Pi računalo moći prepoznati da li se na tom sučelju očekuje izlaz ili ulaz podataka.

```
#GAS SENSOR GPIO
GAS_GPIO = 17
GPIO.setup(GAS_GPIO, GPIO.IN)
print('*****')
print('Inicijalizacija senzor plina')
print('GPIO ==> {0:5d}'.format(GAS_GPIO))
print('*****')
```

Nakon inicijalizacije svih potrebnih senzora za svaki od njih definirana je metoda koja će biti pozvana u trenutku kada se na željenom ulazu pojavi promjena stanja. Metoda je zadužena za definiranje poruke za administratora sustava, upis podataka u Redis i MySQL bazu i slanje podataka prema web aplikaciji putem REST servisa.

```
#CALLBACK FUNCTION FOR GAS SENSOR, id = 6
def gas_sensor_callback(GAS_GPIO):
    poruka = random.choice(gasMessages)
    print('\nSenzor plina: ' + poruka)
    redis_conn.writeToDb('Senzor_plin',
    jsonBuilder.createRedisEntry(6,poruka))
    mysql_conn.insertAlarm(6, poruka)
    payload = jsonBuilder.createPOSTmessage(6, poruka)
    print("Server
response:{0}") .format(restClient.sendPOSTMessage(payload))
```

Osim metoda za obradu podataka pristiglih sa senzora, program ima i metodu koja osluškuje pojavu sistemskog prekida i na njega odgovara zadanim akcijama poput zatvaranja konekcija prema bazi.

```
#CALLBACK FUNCTION FOR KEYBOARD INTERRUPTION
def keyboardInterruptHandler(signal, frame):
    print('\nPrekid sa tipkovnice')
    mysql_conn.closeConnection()
    print('\nBye bye ID:{}'.format(signal))
    GPIO.cleanup()
    sys.exit(0)
```

Nakon metoda definirani su slušači na odgovarajućim pinovima GPIO sučelja koji su definirani metodom osluškivanja (da li se radi podizanju signala na određenom pinu, izostanku signala ili možda oboje), vrijeme nakon kojega metoda može biti ponovo pozvana i naziv same metode.

```
#DETECT GAS
GPIO.add_event_detect(GAS_GPIO, GPIO.RISING, bouncetime=10000)
GPIO.add_event_callback(GAS_GPIO, gas_sensor_callback)
```

U glavnoj metodi programa pozvana je samo metoda za čitanje podataka sa pametnih kartica.

```
if __name__ == "__main__":
    while True:
        try:
            readTag()
        finally:
            pass
```

Nadalje biblioteke koje proširuju mogućnosti glavnog programa kao što su `redisConn.py` ili `mysqlConn.py` sadrže metode kao u nastavku. Metoda `checkConnection()` uspostavlja vezu prema MySQL bazi, odnosno ispisuje grešku ukoliko veza nije uspostavljena.


```

def checkConnection():
    try:
        global connection
        connection = mysql.connector.connect(host=__localhost,
                                             database=__database,
                                             user=__user,
                                             password=__password)

        if(connection.is_connected()):
            print('Uspjesno spajanje na MySQL bazu -
{0}').format(__database)

    except Error as e:
        print('Greska kod spajanja na bazu: {0}').format(e)
        sys.exit(0)

```

Metoda closeConnection() zatvara vezu prema bazi.

```

def closeConnection():
    global connection
    if(connection.is_connected()):
        connection.close()
        print('Zatvorena veza prema MySQL bazi - {0}').format(__database)

```

Metoda insertLog() upisuje zadanu poruku u dnevnik rada aplikacije.

```

def insertLog(message):
    cursor = connection.cursor()
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    cursor.execute("INSERT INTO dnevnik (opis, vrijeme) VALUES
(%s,%s)",(message, timestamp,))
    connection.commit()
    cursor.close()

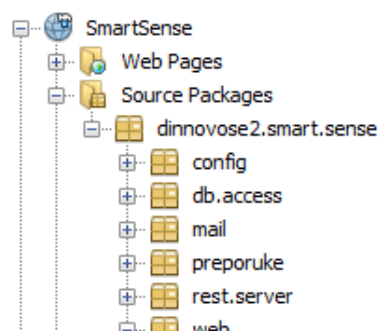
```

Za kraj dočarana je jednostavnost korištenja Python programskog jezika kroz biblioteku `restClient.py` koja sadrži jednu metodu `sendPOSTmessage()` i omogućuje slanje poruke POST metodom na web aplikaciju korisnika.

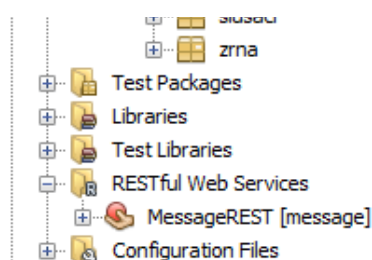
```
url = "http://192.168.0.27:8084/SmartSense/webresources/message"
def sendPOSTMessage(message):
    try:
        r = requests.post(url, message)
        return r
    except:
        print("Neuspjesno slanje POST-a")
```

6.4.4. Programsko rješenje – Java

Web aplikacija pisana u Java programskom jeziku sastoji se od ukupno 34 klase raspoređene u 6 osnovnih paketa. Strukturu aplikacije moguće je vidjeti na dijagramu klasa na Slika 54. Aplikacija raspolaže sa ukupno 8 pogleda koji su temelj korisničkog sučelja ove aplikacije, dok su ostale klase namijenjene kreiranju objekata prema modelima iz baze podataka ili sadrže programsku logiku za obradu tih podataka. Kako bi se što jednostavnije upoznao sa osnovnim konceptima aplikacije u nastavku teksta proći će se po paketnoj strukturi projekta i objasniti najvažnije klase i funkcije koje obavljaju unutar ovog programskog rješenja.

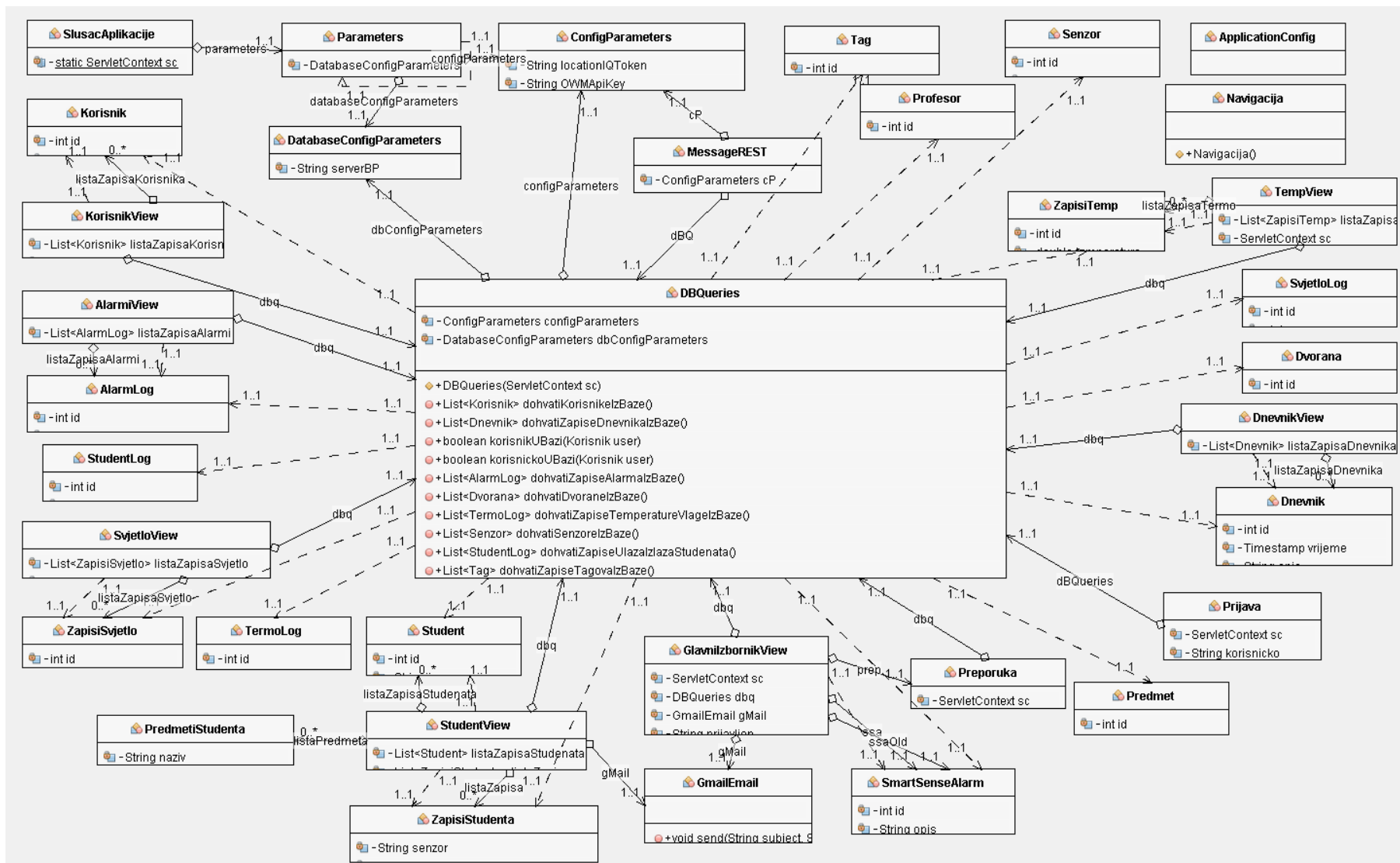


Slika 53: Struktura direktorija Java aplikacije



Klase iz paketa `config` namijenjene su konfiguracijskim parametrima koji se učitavaju po inicijalizaciji konteksta aplikacije. Konfiguracijski parametri su najčešće podaci za spajanje na bazu podataka, serijski brojevi za dohvaćanje podataka sa vanjskih servisa ili jednostavno neke konstante koje se koriste unutar aplikacije. Nadalje unutar paketa `db.access` nalazi se klasa koja objedinjava sve metode potrebne za postavljanje upita na bazu. Paket `mail` sadrži klasu koja programskom rješenju omogućuje slanje e-mail poruka putem Gmail poslužitelja.

Paket `preporuka` implementira klasu koja ovisno o ulaznim podacima alarma koje program dobiva putem REST servisa od Python aplikacije generira adekvatnu akciju odnosno prijedlog koji je kasnije moguće vidjeti na glavnom izborniku korisničkog sučelja aplikacije. Paket `rest.server` kao što mu i samo ime govori implementira poslužitelja REST servisa sa samo jednom metodom koja prima POST zahtjev i prema klijentu šalje adekvatan odgovor. Paket `web` sastoji se od tri dodatna paketa, a zajedno objedinjavaju skup klasa koje čine osnovu web aplikacije. Za svaki od korisničkih pogleda u ovom paketu naći će se odgovarajuće Java zrna i klase sa podacima odnosno programskom logikom potrebnom za ispravno kreiranje odgovora na interakciju korisnika na korisničkom sučelju.



Slika 54: Dijagram klasa Java aplikacije

6.5. Grafičko sučelje

Grafičko sučelje sastoji se od ukupno 8 pogleda preko kojih korisnik ostvaruje interakciju sa aplikacijom. Prvi pogled sa kojim se korisnik susreće je forma za prijavu. Ne razlikuje se od klasičnih formi viđenih na mnogim drugim aplikacijama, a sastoji se od polja za unos korisničkog imena, lozinke, gumba za potvrdu i polja za ispis greške ukoliko korisnik ne postoji ili je upisao pogrešne podatke za prijavu.

Prijava u SmartSense

Korisničko ime	<input type="text" value="korisničko"/>
Lozinka	<input type="text" value="lozinka"/>
	<input type="button" value="Prijava"/>

Slika 55: Forma za prijavu

Nakon uspješne prijave korisnika se preusmjerava na glavni izbornik aplikacije koji se sastoji od izbornika za navigaciju, te prozora sa zadnjim dohvaćenim alarmom i detaljnim opisom imena alarma, prostorije u kojoj je alarm zabilježen, porukom sa alarma i vremenom. Ispod se nalazi predložena akcija ovisno na vrstu alarma. U slučaju kao na Slika 56 gdje se dogodio alarm povišene temperature u prostoru, dohvaćaju se podaci u realnom vremenu sa eksternog servisa, nakon čega se uspoređuju sa vrijednostima u alarmu, te se na temelju usporedbe vrijednosti daje adekvatna preporuka.

SmartSense - Glavni izbornik

The screenshot shows the main menu of the SmartSense application. On the left is a sidebar menu with options: 'Sakrij', 'Izbornik', 'Glavni izbornik', 'Dnevnik', 'Alarmi', 'Korisnici', 'Studenti', 'Temperatura/Vlaga', 'Izlaz iz aplikacije', 'dinnovose2', and 'Odjava'. The main content area displays an alarm notification: 'Alarm' with the message 'Senzor: DHT11 u dvorani: Dvorana 1, poruka: Hmm vruce ili hladno?!' and the time 'Vrijeme: 2020-05-15 16:01:17.0'. Below this, a 'Predložena akcija' (Suggested action) box contains the text: 'T: 31°C, V: 29 %, T vani: 13,94 °C, V vani: 76%' and 'Poruka: Otvoriti prozore da se prostorija prozrači, upaliti odvlaživač zraka'.

Slika 56: Glavni izbornik

Na zaslону navigacije odmah nakon pogleda glavnog izbornika nalazi se pogled dnevnika. Pogled dnevnika sadrži popis svih promjena koje su se dogodile u sustavu, nebitno da li se radi o akcijama iz korisničke ili pozadinske aplikacije sa Raspberry Pi računala. Ovaj pogled ima mogućnost filtriranja zapisa prema opisu ili vremenu događaja.

SmartSense - Dnevnik

Sakrij ▾

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi
- Korisnici
- Studenti
- Temperatura/Vlaga
- Svjetlo
- Izlaz iz aplikacije**
- dinnovose2
- Odjava

ID	Opis	Vrijeme
21	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:05:41.0
22	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:12:57.0
23	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:13:39.0
24	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:14:46.0
25	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:20:49.0
26	Senzor: MQ2 - Detektor plina, Poruka: Osjeti se plin!	2020-05-05 18:25:00.0
27	Senzor: MQ2 - Detektor plina, Poruka: Osjeti se plin!	2020-05-05 18:26:23.0
28	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:57:39.0
29	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:57:44.0
30	Senzor: DHT11 - Senzor temperature i vlage, Poruka: Izmjerena temperatura	2020-05-05 18:57:48.0

Slika 57: Pogled dnevnik

Nakon pogleda dnevnika dolazi se na pogled alarma. On se od zapisa dnevnika razlikuje po tome što prikazuje samo akcije iz pozadinske aplikacije sa Raspberry Pi računala koja detektira promjene na sensorima. Ovisno o stanju na senzoru i zadovoljenom uvjetu odnosno detektiranom alarmu podaci se zapisuju u bazu i prikazuju kronološki na ovom pogledu. Pogled također ima mogućnosti filtriranja prema poruci alarma i vremenu događaja.

SmartSense - Alarmi

Sakrij ▾

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi**
- Korisnici
- Studenti
- Temperatura/Vlaga
- Izlaz iz aplikacije
- dinnovose2
- Odjava

ID	Senzor id	Poruka	Vrijeme
321	8	Ovdje nesto gori!?!?	2020-05-15 11:23:13.0
322	7	V O D A detektirana	2020-05-15 11:24:59.0
323	4	Nesto se mice!	2020-05-15 11:25:04.0
324	3	Ovdje je pao mrak?!?	2020-05-15 11:32:02.0
325	4	Vidim kretanje!	2020-05-15 11:32:13.0
326	8	V A T R A detektirana	2020-05-15 11:34:14.0
327	4	Nesto se mice!	2020-05-15 11:34:15.0
328	4	Vidim kretanje!	2020-05-15 11:40:30.0
329	4	Kretanje u dvorani detektirano	2020-05-15 11:40:47.0
330	4	Nesto se mice!	2020-05-15 11:41:02.0
331	4	Nesto se mice!	2020-05-15 11:41:14.0

Slika 58: Pogled alarmi

Pogled korisnika sadrži popis svih korisnika i njihovih e-mail adresa odnosno korisničkih imena s kojima je moguće napraviti prijavu u aplikaciju.

SmartSense - Korisnici

Sakrij

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi
- Korisnici
- Studenti
- Temperatura/Vlaga
- Svjetlo
- Izlaz iz aplikacije**
- dinnovose2
- Odjava

ID	Ime	Prezime	Email	Korisničko
1	Dino	Novosel	dinnovose2@foi.hr	dinnovose2
2	Ivan	Marinkovic	imar@email.com	imar12
3	Marko	Ivancic	mariv@email.com	marivan45
4	Lucija	Marulic	lucmar@email.com	lucimare96
5	Karla	Kincl	karlkin@email.com	karlkin456

Slika 59: Pogled korisnici

Pogled sa studentima nešto je kompliciraniji jer omogućuje odabir studenta iz padajućeg izbornika nakon čega se pritiskom na gumb filtriraj dohvaćaju zapisi kretanja po dvoranama ustanove. Prema dohvaćenim zapisima kretanja izračunava se provedeno vrijeme na nastavi prema rasporedu predavanja te se u drugoj tablici ispisuju svi predmeti odabranog studenta sa ukupnim brojem odslušanih sati. Sa ovog pogleda moguće je studentu poslati e-mail poruku sa sažetkom o stanju odslušanih sati za sve predmete.

SmartSense - Studenti

Sakrij

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi
- Korisnici
- Studenti
- Temperatura/Vlaga
- Svjetlo
- Izlaz iz aplikacije**
- dinnovose2
- Odjava

Student: Dino Novosel Filtriraj zapise studenta **Filtriraj** Obavijest mailom **Pošalji**

Senzor	Dvorana	Vrijeme ulaska	Vrijeme izlaska	Proteklo vrijeme
RFID	Dvorana 1	2020-05-12 15:06:00.0	2020-05-12 18:52:51.0	03:46:51
RFID	Dvorana 1	2020-05-17 20:12:49.0	2020-05-17 20:27:01.0	00:14:12
RFID	Dvorana 1	2020-05-17 20:27:01.0	2020-05-17 20:28:56.0	00:01:55
RFID	Dvorana 1	2020-05-17 20:30:51.0	2020-05-17 20:32:28.0	00:01:37

Predmet	Sati	Uvjet	Odslusano
Matematika	60	50	4.4
Statistika	60	50	2.4
Napredno programiranje	60	50	1.5

Slika 60: Pogled studenti

Nakon pogleda sa studentima u aplikaciji dolazi se do pogleda sa zapisima temperature i vlage. Kako je prvobitna ideja bila napraviti filtriranje po prostorijama, a dostupan je bio samo jedan senzor vlage i temperature tako i je forma ostala bez padajućeg izbornika, umjesto kojeg su direktno na tablicu dodana tekstualna polja za detaljnije filtriranje zapisa.

SmartSense - Dnevnik temperature i vlage

Sakrij

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi
- Korisnici
- Studenti
- Temperatura/Vlaga
- Izlaz iz aplikacije**
- dinnovose2
- Odjava

ID	Senzor	Dvorana	Temperatura	Vlaga	Vrijeme
81	DHT11	Dvorana 1	24.0 °C	46.0 %	2020-05-12 12:39:49.0
82	DHT11	Dvorana 1	24.0 °C	46.0 %	2020-05-12 12:44:50.0
83	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 12:54:52.0
84	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 12:59:53.0
85	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:04:54.0
86	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:05:54.0
87	DHT11	Dvorana 1	24.0 °C	46.0 %	2020-05-12 13:10:55.0
88	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:20:57.0
89	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:25:58.0
90	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:30:59.0
91	DHT11	Dvorana 1	24.0 °C	47.0 %	2020-05-12 13:41:01.0

Slika 61: Pogled dnevnika temperature i vlage

Zadnji pogled ove web aplikacije vezan je za zapise potrošnje električne energije. Na ovom pogledu odnosno formi korisnik ima mogućnost upisa snage rasvjetnih tijela u prostoriji, a sustav će mu na osnovu proteklog vremena izračunati vrijeme i okvirnu cijenu ukupno potrošene električne energije.

SmartSense - Dnevnik svjetla

Sakrij

Izbornik

- Glavni izbornik
- Dnevnik
- Alarmi
- Korisnici
- Studenti
- Temperatura/Vlaga
- Svjetlo
- Izlaz iz aplikacije**
- dinnovose2
- Odjava

Izračun potrošnje: Snaga rasvjete (W)

Ukupno sati 185 Cijena utrošene energije: 1155.0 kn

ID	Senzor	Dvorana	Vrijeme paljenja	Vrijeme gašenja	Protetklo vrijeme
1	HC-SR501	Dvorana 2	2020-05-12 15:06:00.0	2020-05-12 18:52:51.0	03:46:51
2	HC-SR501	Dvorana 2	2020-05-13 15:06:00.0	2020-05-13 19:52:51.0	04:46:51
3	HC-SR501	Dvorana 2	2020-05-14 10:06:00.0	2020-05-14 12:52:51.0	02:46:51
4	HC-SR501	Dvorana 2	2020-05-14 14:06:00.0	2020-05-14 16:52:51.0	02:46:51
5	HC-SR501	Dvorana 2	2020-05-15 11:06:00.0	2020-05-15 12:52:51.0	01:46:51
6	HC-SR501	Dvorana 2	2020-05-15 14:06:00.0	2020-05-15 18:52:51.0	04:46:51

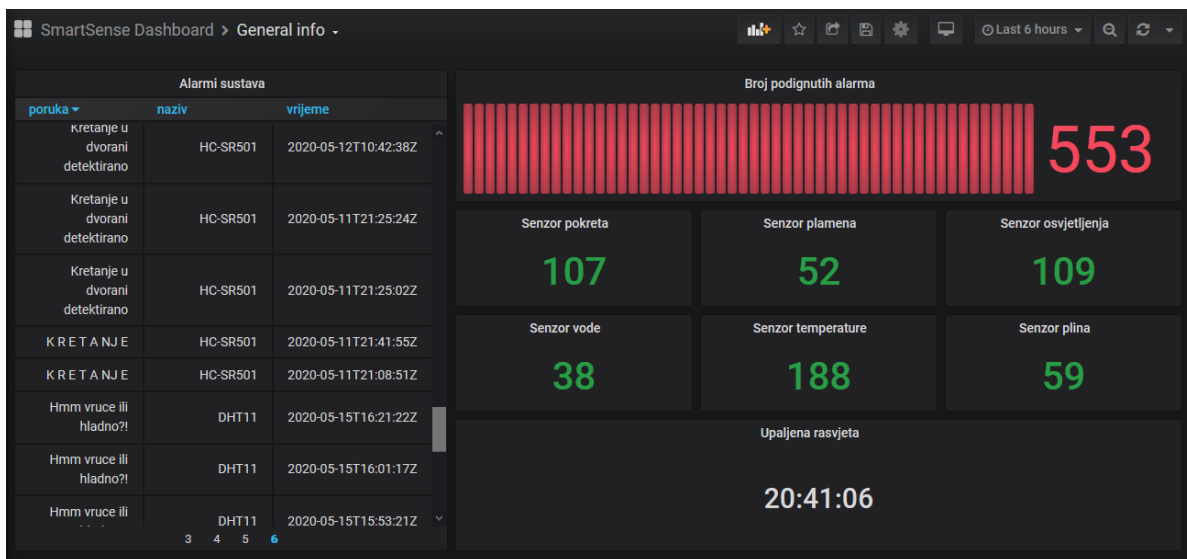
Slika 62: Pogled dnevnika svjetla

Svi pogledi na kraju izbornika imaju zapis o trenutno prijavljenom korisniku i gumbom za odjavu iz aplikacije. Pritiskom na gumb korisnik se ponovo preusmjerava na formu za prijavu.

6.6. Dodatne mogućnosti - Grafana

Grafana je programski paket otvorenog koda za analitiku i grafički prikaz podataka. Glavni građevni elementi jednog pogleda u Grafani su grafikoni i tekstualne vizualizacije iz raznih skladišta podataka kao što su Prometheus, MySQL, PostgreSQL, InfluxDB. Pogledi omogućuju jednostavan pregled i postavljanje graničnih vrijednosti za svaki od parametara koji se nadgleda. Zajednička komponenta svim pogledima je raspon vremena za koji se promatraju dohvaćene vrijednosti. Razna proširenja programskog paketa dostupna su putem službenog repozitorija stoga je ograničavajući faktor sam korisnik i njegova potreba za finijom odnosno grubljom granulacijom prikaza podataka.

Za potrebe projekta izrađena su tri pogleda koji na jednostavan i pregledan način opisuju stvarno stanje sustava. Prvi pogled pod nazivom „General info“ prikazuje opće stanje sustava odnosno listu sa svim podignutim alarmima, ukupan broj alarma te broj alarma po pojedinoj kategoriji. Ispod podataka o alarmima nalazi se brojač koji pokazuje ukupan broj sati upaljene rasvjete.



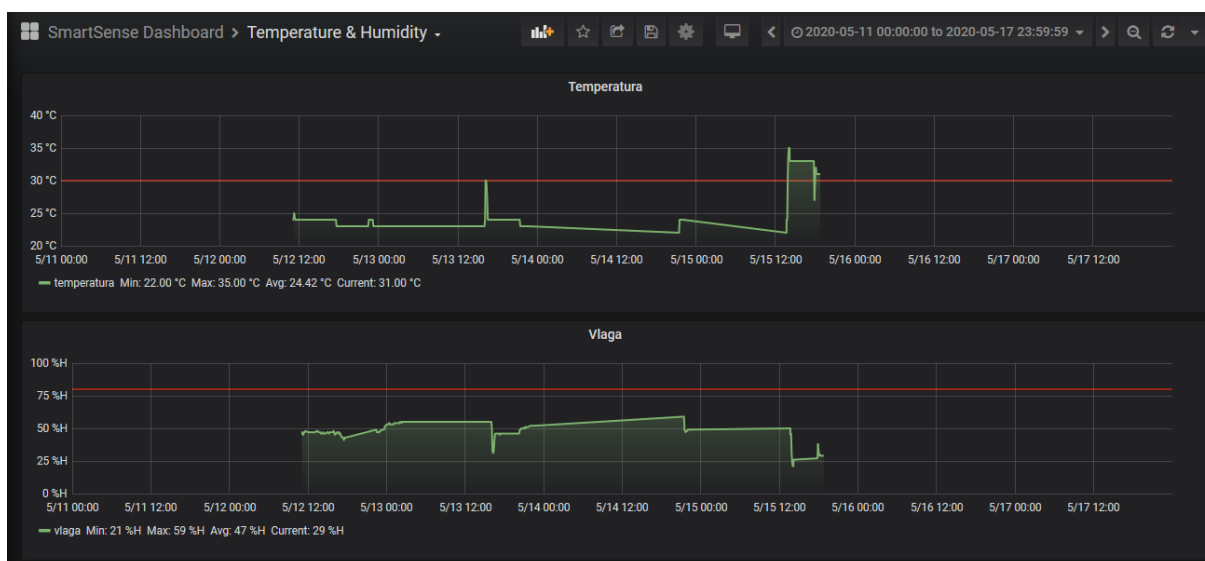
Slika 63: Grafana – pregled sustava

Drugi pogled pod nazivom „Alarms“ sadrži varijablu sa spremjenim svim identifikacijskim oznakama senzora iz sustava. Odabirom jedne od oznaka na grafikonu može se filtrirati i prikazati alarme odabranog senzora. Osim vremena u kojem se alarm dogodio na grafikonu može se pronaći i poruka sustava koju je prosljedio na grafičko sučelje Java web aplikacije.



Slika 64: Grafana - alarmi u sustavu prema ID

Treći pogled po imenu „Temperature & Humidity“ prikazuje grafikone temperature i vlage. Na pogledu su definirane i gornje granice koje označavaju razinu nakon koje sustav sam automatski podiže interni alarm. Uvjet za podizanje alarma je prekoračenje srednje vrijednosti iznosa temperature i vlage u zadnjih 5 minuta. Srednja vrijednost za potrebe ovog projekta postavljena je na 30°C za temperaturu, dok je za vlagu vrijednost postavljena na 80%. Ako je uvjet zadovoljen sustav sprema alarm u interni dnevnik i šalje obavijest e-mail porukom administratoru sustava.



Slika 65: Grafana - temperatura i vlaga

7. Prednosti i nedostaci Raspberry Pi računala

Nakon dugog i intenzivnog druženja sa Raspberry Pi računalom moguće je doći do nekih zaključaka. Prvi je definitivno taj će Vas ovo jednostavno računalo na prvu osvojiti i da će pozitivnih reakcija biti puno više od onih negativnih. Zamjerke će brzo biti zaboravljene i ostavljene po strani, dok smišljate nove načine kako ih zaobići i dodatno se uveseliti nakon što u tome doista i uspijete.

7.1. Prednosti Raspberry Pi računala

Nezaobilazna točka koja ujedno predstavlja i najveći marketinški podvig ovog računala je njegova cijena. Teško da će korisnik za slične novce danas na tržištu pronaći tako svestranu platformu koja na jednostavan način objedinjuje sve karakteristike jednog modernog računala i mikroupravljača. Spajanje senzora i drugih elektroničkih uređaja nikada nije bilo jednostavnije, a komunikacija s njima brža. Kako ne bih odmah na početku krenuli u neke dublje analize, na početku uz cijenu bitno je naglasiti i veličinu samog računala. Veličina nešto veća od standardne kreditne kartice čini ovaj uređaj izuzetno prenosivim. Ako se u igru uključe i na tržištu dostupna kućišta koja povećavaju sigurnost računala tijekom transporta dolazi se do dobitne kombinacije. Gotovo u svakom trenutku kada pri ruci postoji nekakav monitor ili TV uređaj korisnik ima spremno gotovo računalo. Mogućnosti personalizacije različitom periferijom zaista su nebrojene. Raspberry Pi podržava gotovo sve vrste tipkovnica, miševa, printera, mrežnih kartica, projektora, kamera i sl, a same potrebe korisnika određuju granice iskoristivosti.

Idealna je platforma za učenje. Bilo da se radi o programskim jezicima, upoznavanjem sa raznim operacijskim sustavima ili samo istraživanju mogućnosti. Svatko će pronaći nešto za sebe. Primarna okolina temeljena na Linux-u garantira veliku količinu programskih alata, a priroda otvorenog koda osigurava mnoge načine personalizacije. Eksperimentiranje na ovom računalu definitivno je imperativ. Brza izmjena memorijskih kartica omogućuje promjene radne okoline u nekoliko trenutaka, a kompatibilnost sa drugim Raspberry Pi uređajima osigurava prenosivost i ponovnu iskoristivost. Neki od aduta novijih Raspberry Pi računala četvrte generacije je ugrađena podrška za prikaz na više monitora i prikaz videa u visokoj kvaliteti sve do 4K razlučivosti.

7.2. Nedostatci Raspberry Pi računala

Gotovo svaki korisnik koji je imao doticaja sa Raspberry Pi serijom računala kao prvi nedostatak navesti će slab procesorski i memorijski potencijal. Ovakva osuda dolazi najprije zbog navike korisnika da riječ računalo shvaćaju kao stroj velike procesorske i memorijske snage namijenjenog za obavljanje uredskog posla, igranje igara i sl. Danas kada su gotovo svi mobilni uređaji srednje klase na tržištu sklopovski jači od Raspberry Pi računala ova zamjerka donekle i drži vodu. Tom problemu izdavanjem zadnje verzije Raspberry Pi uređaja pokušali su doskočiti i sami proizvođači tako što su u ponudu uvrstili i računalo sa 4GB radne memorije. Koliko će im to uspjeha donijeti pokazati će vrijeme, obzirom da je i cijena samog uređaja znatno porasla. Kada smo već kod cijene treba napomenuti da Raspberry Pi računalo gotovo uvijek dolazi isporučeno bez periferije za rad, počevši od napajanja pa sve do komponenti poput HDMI konektora smanjenih dimenzija i sl. Zbog toga je korisnik primoran uložiti dodatne novce kako bi uopće mogao raditi na računalu.

Još jedan od nedostataka tiče se kompatibilnosti odnosno nedostatka istoga sa Windows platformom. U današnje vrijeme kada na svijetu postoji otprilike 1.5 milijardi korisnika Windows platforme postoji veliki financijski, ali i poduzetnički potencijal kako pridobiti još aktivnih korisnika. Do sada je postojao samo jedan projekt pod imenom Windows IoT Core koji je poslužio da se Windows platforma približi korisnicima Raspberry Pi računala. Ovaj projekt koristio je Raspberry Pi računalo samo kao posrednika za komunikaciju između vanjskih senzora i korisničkog računala, te nije u potpunosti iskoristavao sve mogućnosti koje Raspberry Pi platforma pruža. Tijekom izrade ovoga rada bilo je pokušaja instalacije Windows IoT Core sustava, ali bezuspješno. Dokumentacija čitavog projekta dosta je šturo napisana, a sam projekt izgleda kao neuspješni pokušaj implementacije donekle dobre ideje. Za kraj ono što će najviše mučiti korisnika Raspberry Pi računala definitivno je pregrijavanje. Naime računalo ne dolazi niti sa pasivnim niti aktivnim hladnjacima, pa će svako malo dolaziti do povećanja temperature na procesoru i pada performansi. Stoga je neko ozbiljnije opterećenje sustava gotovo nemoguće izvesti bez adekvatnog hlađenja.

8. Zaključak

Raspberry Pi idealna je platforma za ulazak u svijet računarstva. Pruža jedinstvenu mogućnost spajanja teoretskih i praktičnih znanja na vrlo jednostavan i zabavan način. Osnovno predznanje za korištenje Raspberry Pi računala gotovo da nije niti potrebno. Količina dostupne dokumentacije i velika zajednica korisnika osigurati će tražene informacije čak i najzahtjevnijim korisnicima. Mlađa populacija također neće biti izostavljena obzirom na činjenicu da službeni operacijski sustav dolazi sa instaliranim programskim alatima namijenjenima baš za njih, dok će napredni korisnici kroz podršku raznim tehnologijama iznova pronalaziti načine za stvaranje inovacija. Mogućnost objedinjavanja sklopovskih komponenti i programske potpore na ovoliko jednostavan način gotovo je jedinstvena na tržištu, a zahvaljujući implementaciji ulaza i izlaza opće namijene na tiskanu pločicu ovaj jednostavni komad sklopovlja moguće je dodavanjem jeftine elektronike pretvoriti u vrlo složen sustav.

Iako procesorskom i memorijskom snagom ne može u potpunosti zamijeniti stolna računala, jer u konačnici za to nije niti namijenjeno, većina korisnika neće ostati ravnodušna mogućnostima i funkcionalnostima ovog proizvoda. Koliki je zapravo obrazovni potencijal Raspberry Pi računala može se pretpostaviti već po pogledu na pred instalirane programske pakete koji se nalaze na službenom operacijskom sustavu. Raspbian operacijski sustav dolazi standardno sa podrškom za C, C++, Scratch, Ruby, Python I Java tehnologije, kao i sa desetak integriranih razvojnih okruženja koji omogućavaju jednostavan razvoj programskih rješenja. Neke od funkcionalnosti dostupnih tehnologija moguće je vidjeti i iz programskog rješenja implementiranog u svrhu izrade ovog rada, koje na vrlo jednostavan način korištenjem vanjskih senzora putem GPIO sučelja i Python programskog jezika omogućuje komunikaciju između sklopovlja i programske potpore. Korištenje Docker sustava virtualizacije omogućilo je korištenje standardnih servisa na još fleksibilniji i jednostavniji način, a implementacija web sučelja korisniku je omogućila jednostavnu interakciju sa cjelokupnim sustavom.

Daljnji napredak tehnologije i implementacija jačih sklopovskih komponenti u budućnosti će ovo računalo učiniti još svestranijim. Tako da će obrazovni element korištenja još više rasti, a mogućnosti postati nebrojene. Kao i do sada motivacija i želja za znanjem biti će jedina prepreka korisniku da istraži sve mogućnosti, i da ovo računalo iskoristi na sebi svojstven način.

Popis literature

- [1] Raspberry Pi Foundation , Raspberry Pi 3 Model B, 2020.
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (dostupno 21.04.2020.)
- [2] Raspberry Pi Foundation, *Raspberry Pi documentation*, 2020.
<https://www.raspberrypi.org/documentation/> (dostupno 21.04.2020)
- [3] S. Monk, *Raspberry Pi cookbook*. Beijing: O'Reilly, 2014.
- [4] W. Gay, *Mastering the Raspberry Pi*. 2014.
- [5] "ARM Cortex-A53 MPCore Processor Technical Reference Manual", 2016.
- [6] A. Pajankar, "Instant Raspberry Pi," 2016.
- [7] G. Halfacree, *Raspberry Pi Beginner's Guide*, 2019.
- [8] P. Cobbaut, *Linux Fundamentals*. 2015.
- [9] M. B. S. Zalavadia, *Linux Architecture and Advantages*, 2016.
- [10] M. Garrels, *Introduction to Linux*, 2008.
- [11] D. J. Barrett, R. E. Silverman, *SSH, the secure shell: the definitive guide*, 2001.
- [12] C. Chung, *SSH File Transfer Protocol (SFTP) Explained*, 2014.
- [13] Marc Wilson, *SCP – What is Secure Copy Protocol – Definition & Example*, 2019.
<https://www.pcwdld.com/what-is-scp> (dostupno 21.04.2020)
- [14] *VNC® User Guide*. RealVNC Limited, 2015.
- [15] J. Leonardo, K. Milan, C. Dario, *Sustavska programska potpora*, 2010.
- [16] Raspbian, 2020. <https://www.raspberrypi.org/documentation/raspbian/> (dostupno 21.04.2020.)
- [17] Ubuntu MATE for Raspberry Pi, 2020. <https://ubuntu-mate.org/ports/raspberry-pi/> (dostupno 21.04.2020.)
- [18] Ubuntu Server for ARM, 2020. <https://ubuntu.com/download/server/arm> (dostupno 21.04.2020.)
- [19] Introduction to RISC OS, 2020.
<https://www.riscosopen.org/wiki/documentation/show/Introduction%20to%20RISC%20OS> (dostupno 21.04.2020.)
- [20] BerryBoot v2.0 – bootloader, 2020.
<https://www.berryterminal.com/doku.php/berryboot> (dostupno 21.04.2020.)
- [21] J. Mladen, *Programski jezik C*, 2004.
- [22] S. Fajković, *Uvod u programiranje*, 2015.

- [23] R. Blum and C. Bresnahan, *Sams teach yourself Python programming for Raspberry Pi in 24 hours*. Indianapolis, Indiana: Sams, 2014.
- [24] M. Čupić, *Programiranje u Javi*, 2015.
- [25] BlueJ documentation, 2020. <https://www.bluej.org/doc/documentation.html> (dostupno 21.04.2020.)
- [26] T. Enrico, T.Nick, *Geany*, 2019. <https://www.geany.org/manual/current/index.html> (dostupno 21.04.2020.)
- [27] Scratch Wiki, 2020. https://en.scratch-wiki.info/wiki/Scratch_Wiki (dostupno 21.04.2020.)
- [28] Mouser Electronics, "DHT11 Humidity & Temperature Sensor"
- [29] Waveshare, „MQ-2 Gas Sensor User Manual“
- [30] "MFRC522 Standard performance MIFARE and NTAG frontend,", 2016.
- [31] "GL55 Series Photoresistor Manual"
- [32] A. Lady, "PIR Motion Sensor", 2020.
- [33] Handson Technology, "2 Channel 5V Optical Isolated Relay Module"
- [34] D. Vrčić, *Uvod u Docker*, 2016.
- [35] "MySQL Workbench reference manual", 2020.

Popis slika

Slika 1: Shematski prikaz važnijih komponenti Raspberry Pi računala (Prema: Liz Upton, 2011, dostupno na: https://www.raspberrypi.org/blog/new-graphic/).....	5
Slika 2: Specijalizirana Raspberry Pi periferija - ekran osjetljiv na dodir i kamera (Prema: http://thepihut.com).....	10
Slika 3: Struktura direktorija Linux operacijskog sustava (Dostupno na: http://www.linuxandubuntu.com/home/the-linux-file-system-structure-explained)	14
Slika 4: Naredba clear	16
Slika 5: Naredba cd	16
Slika 6: Naredba mkdir	17
Slika 7: Naredba ls	17
Slika 8: Naredba touch	17
Slika 9: Naredba ls -l	17
Slika 10: Naredba ls -a	18
Slika 11: Naredba pwd	18
Slika 12: Naredba who	18
Slika 13: Naredba rmdir.....	19
Slika 14: Naredba cat	19
Slika 15: Naredba cp	19
Slika 16: Naredba mv - premještanje.....	19
Slika 17: Naredba mv - preimenovanje.....	20
Slika 18: Naredba stat	20
Slika 19: Naredba man	20
Slika 20: Prijenos podataka SFTP protokolom (prema: John Hughes, <i>An introduction to Secure Shell Access and Secure File Transfer Protocol</i> , 2018)	23
Slika 21: Komunikacija VNC protokolom.....	24
Slika 22: Operacijski sustavi za Raspberry Pi (Dostupno na: https://www.raspberrypi.org/downloads/)	25

Slika 23: Radna površina Raspbian operacijskog sustava	27
Slika 24: Konfiguracija sučelja na Raspbian operacijskom sučelju.....	28
Slika 25: Struktura programskih paketa Ubuntu MATE operacijskog sustava	30
Slika 26: Radna površina RISC operacijskog sustava (Preuzeto sa: https://i.ytimg.com/vi/oL4w3AK6Qpw/maxresdefault.jpg).....	33
Slika 27: Berryboot glavni izbornik.....	35
Slika 28: Programski jezici C++, Java, Python.....	38
Slika 29: Kompiliranje i pokretanje programa pisanog u C-u.....	40
Slika 30: Kompiliranje i pokretanje programa u C++	42
Slika 31: Pokretanja programa pisanog u Python-u	44
Slika 32: Kompiliranje i pokretanje programa u Javi	46
Slika 33: Uređivač teksta na BlueJ okolini	46
Slika 34: Geany razvojno okruženje	48
Slika 35: Glavni zaslon Scratch okruženja.....	49
Slika 36: Model prostora sa sensorima.....	50
Slika 37: DHT11 digitalni senzor temperature i vlage (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/31EM6oovwLL_SX466_.jpg)	54
Slika 38: MQ-2 digitalni senzor plina (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/51S1R01YqUL_SX342_.jpg).....	55
Slika 39: Senzor za kišu (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/61uO4BlnY-L_SX466_.jpg).....	55
Slika 40: Senzor plamena (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/71wv%2BsRpmWL_SL1500_.jpg)	56
Slika 41: RFID senzor sa pametnim karticama (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/61H2HVrbU2L_AC_SX466_.jpg)	57
Slika 42: Fotosjetljivi senzor (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/51OxUXS43FL_SX342_.jpg)	57
Slika 43: Senzor pokreta (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/61KSJvt3JnL_AC_SX466_.jpg)	58

Slika 44: Dvokanalna elektromagnetna sklopka (Preuzeto sa: https://images-na.ssl-images-amazon.com/images/I/51eyw8a9hKL.jpg)	59
Slika 45: Naredba docker images	61
Slika 46: Naredba docker ps.....	62
Slika 47: MySQL Workbench korisničko sučelje	64
Slika 48: NetBeans razvojno okruženje	64
Slika 49: Arhitektura sustava	65
Slika 50: Raspored senzora na GPIO sučelju	66
Slika 51: Redis baza - primjer dohvaćanja podataka	67
Slika 52: Dijagram entiteta i veza MySQL baze	68
Slika 53: Struktura direktorija Java aplikacije.....	73
Slika 54: Dijagram klasa Java aplikacije	75
Slika 55: Forma za prijavu	76
Slika 56: Glavni izbornik	76
Slika 57: Pogled dnevnik	77
Slika 58: Pogled alarmi.....	77
Slika 59: Pogled korisnici.....	78
Slika 60: Pogled studenti	78
Slika 61: Pogled dnevnika temperature i vlage	79
Slika 62: Pogled dnevnika svjetla	79
Slika 63: Grafana – pregled sustava.....	80
Slika 64: Grafana - alarmi u sustavu prema ID	81
Slika 65: Grafana - temperatura i vlaga	81

Popis tablica

Tablica 1: Specifikacije Raspberry Pi 3 Model B računala	4
Tablica 2: Sažetak prednosti i mana Linux operacijskih sustava.....	13
Tablica 3: Osnovni direktoriji na Linux operacijskom sustavu	15
Tablica 4: Omjer temperature i vlage u prostoru (preuzeto sa: https://proluft.hr/savjeti-i-blog/savjeti/kucanstvo/optimalna-temperatura-i-vlaznost-ugodan-boravak-46/).....	52
Tablica 5: Popis tablica MySQL baze	69