

RAZVOJ PROGRAMSKE OPREME

Naloga 3

Table of Contents

Predstavitev članov skupine.....	3
Alex Brence.....	3
Ivor Canjuga.....	4
Matko Zlatič.....	5
Marko Kričej.....	6
Predstavitev ideje tima.....	7
Tehnologije za dejanski razvoj.....	8
Vzpostavitev osnovnega časovnega načrta in natančnih opravil za naslednji teden.....	9
Natančen plan za naslednji teden (8. 11. - 14. 11.).....	11

Predstavitev članov skupine

(Ime priimek, izkušnje (tehnologije, razvoj), interesna področja (tehnologije, razvoj, nasploh računalništvo))

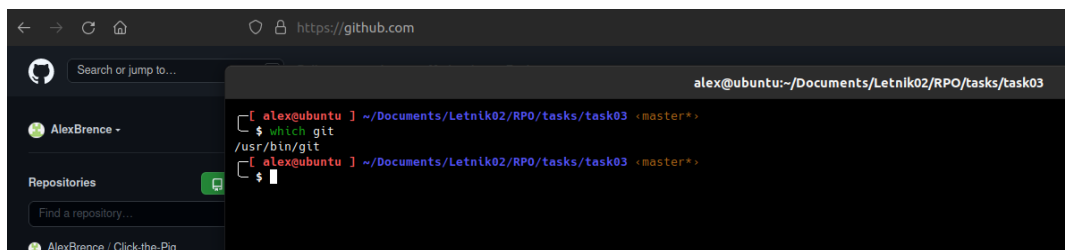
Alex Brence

Izkušnje:

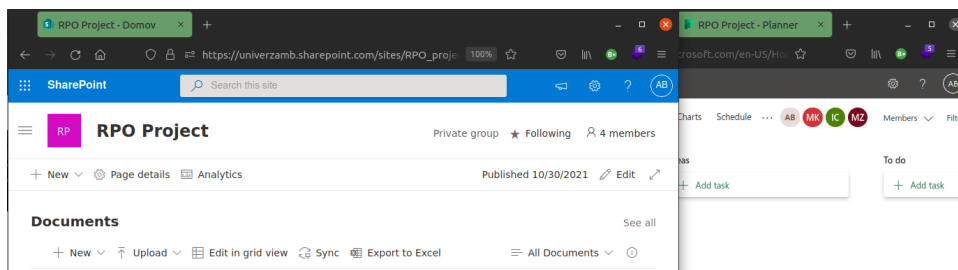
- C++ (konzolne, SFML (osnove), QT (osnove))
- Python (PyGame (osnove), Django)
- Rust (se še učim)
- Hekanje (osnove)
- Linux

Interesna področja:

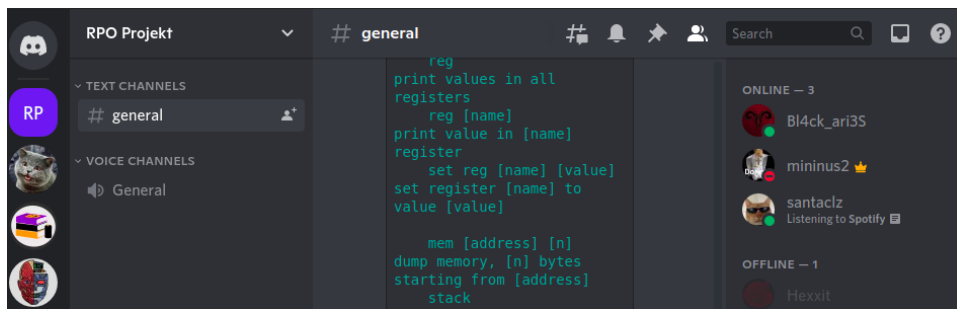
- Desktop aplikacije, Vgrajeni sistemi, Umetna inteligenca, Robotika, Machine learning, hekanje, competitive programming



Github in git



MS Planner in Sharepoint



Discord

Za ne-projektne okolje uporabljam Vim.

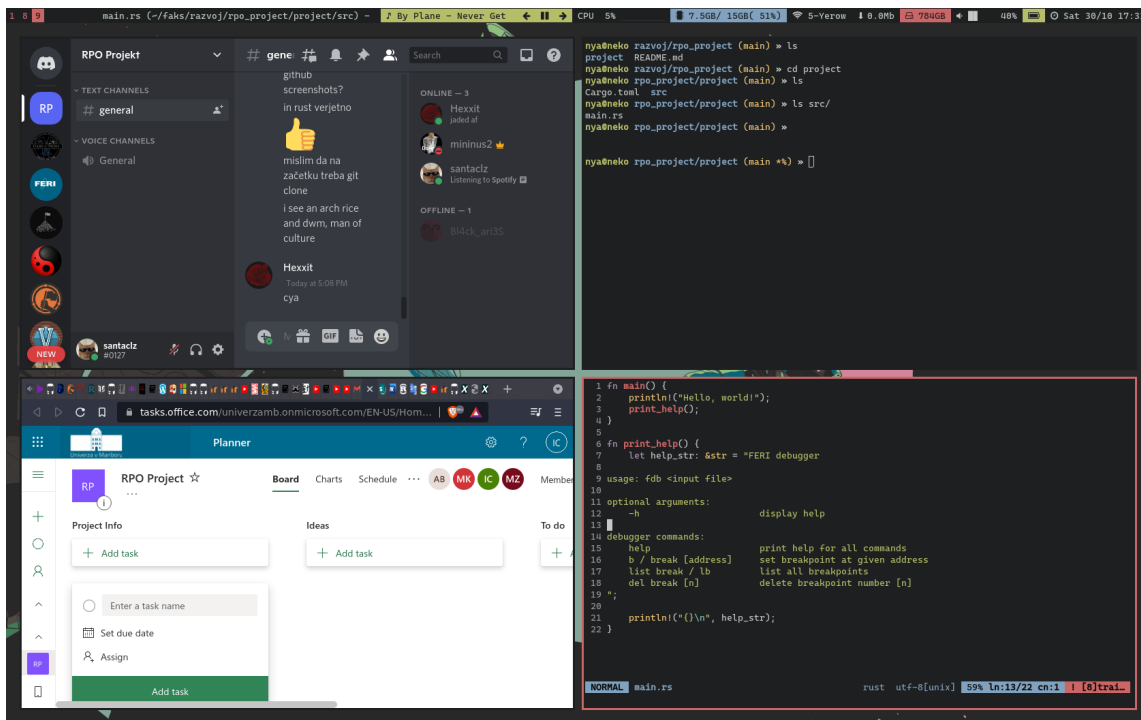
Ivor Canjuga

Izkušnje:

- Jeziki: Python, C, C++, Rust, C#, Java, Bash
- Zbirni jeziki: x86, x86_64, ARM
- Razvijanje iger (Unity, C#)
- Razvijanje prevajalnika (Rust)
- Hekanje – web, kriptografija, reverse engineering, binary exploitation

Interesna področja:

- Iskanje 0day ranljivosti, Operacijski sistemi, Vgrajeni sistemi, Fuzzing, Prevajalniki, Kernel exploitation, Umetna inteligenca



Slika: namestitev okolja, ne-projektne okolje: vim

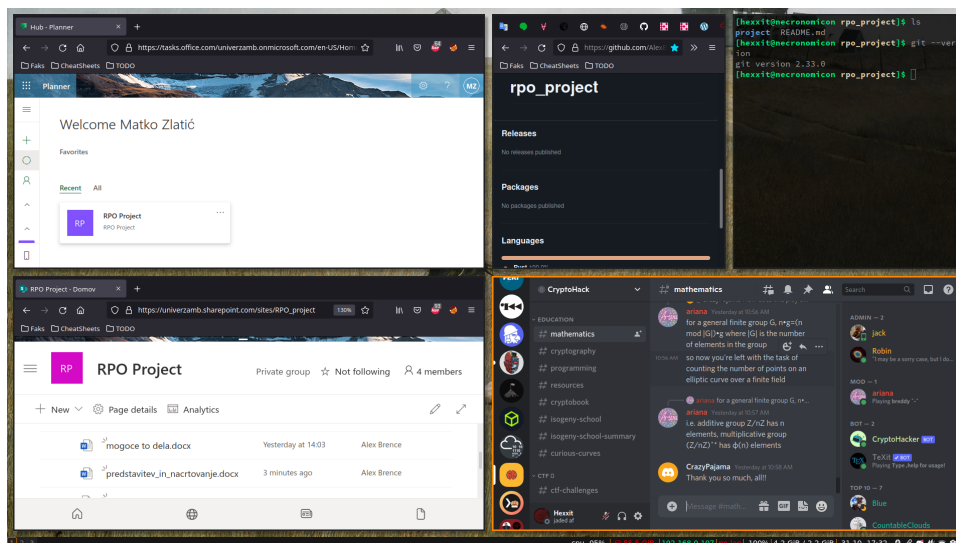
Matko Zlatić

Izkušnje:

- Jeziki: C, C++, C#, Python, Java, JavaScript, Bash, PHP
- Razvijanje iger (GML, Unity, C#)
- Hekanje – web, kriptografija, reverse engineering

Interesna področja:

- Hekanje, kriptografija, vgrajeni sistemi, polimorfizem, tehnologije veriženja blokov



- Ne-projektne okolje: Atom

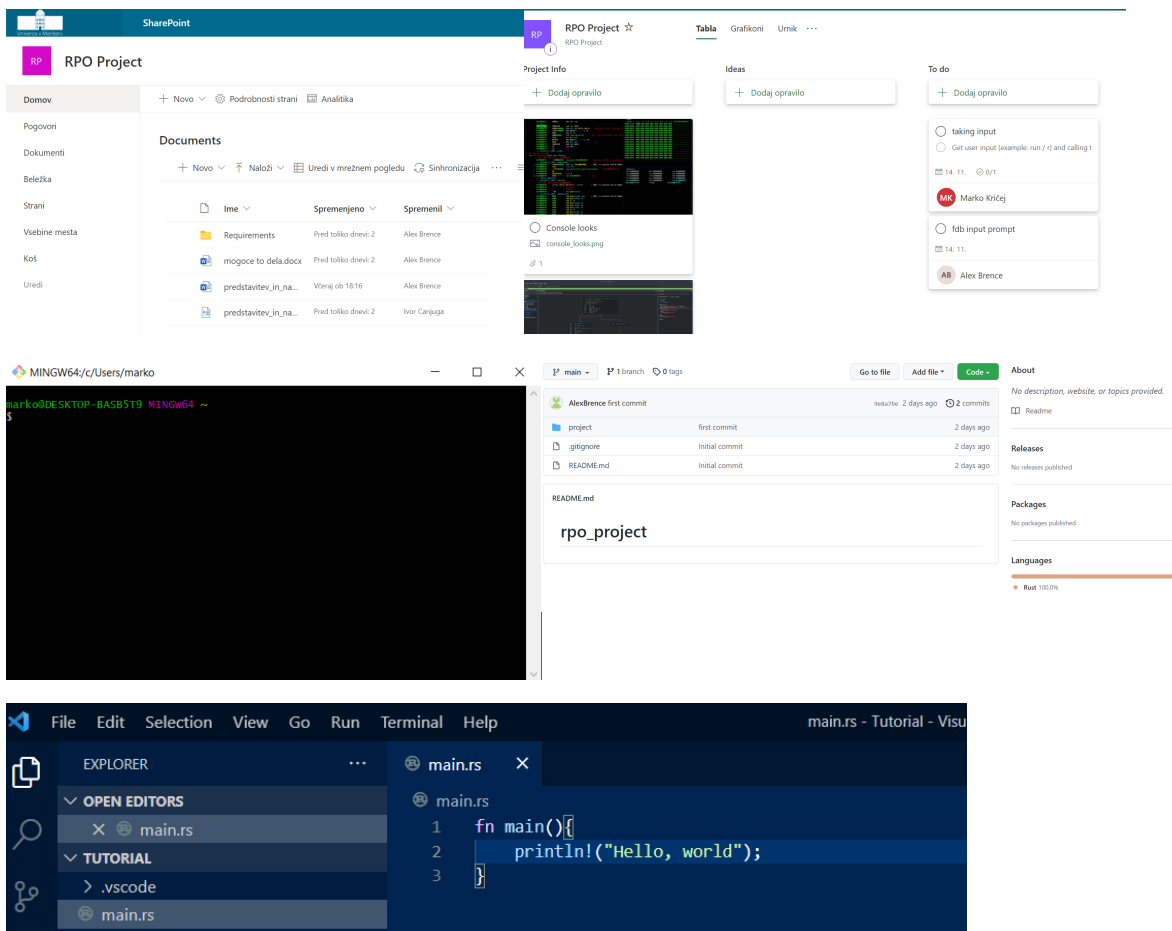
Marko Kričej

Izkušnje:

- Jeziki: C++, Python, Bash
- Tekmovalno programiranje

Interesna področja:

- Umetna inteligenca, strojno učenje, tekmovalno programiranje



Ne-projektno okolje: Visual studio code

Predstavitev ideje tima

FERI Debugger (fdb)



The screenshot shows the radare2 disassembler interface. The main panel displays assembly code for a function named `sym.imp.printf`. The code includes instructions like `mov rbp, rsp`, `sub rsp, 0x10`, `lea rdi, str.Hello_World`, `mov dword [local_4h], 0`, `mov al, 0`, `call sym.imp.printf`, `xor ecx, ecx`, `mov dword [local_8h], eax`, `mov eax, ecx`, `add rsp, 0x10`, and `pop rbp`. The stack panel shows the current stack frame with offsets and values. The registers panel shows the state of various registers, including `rax`, `rbx`, `rcx`, `rdi`, `rsi`, `r8`, `r9`, `r10`, `r11`, `r12`, `r13`, `r14`, `r15`, `rbp`, and `rflags`.

Illustration 1: radare2 (izgled v konzoli)

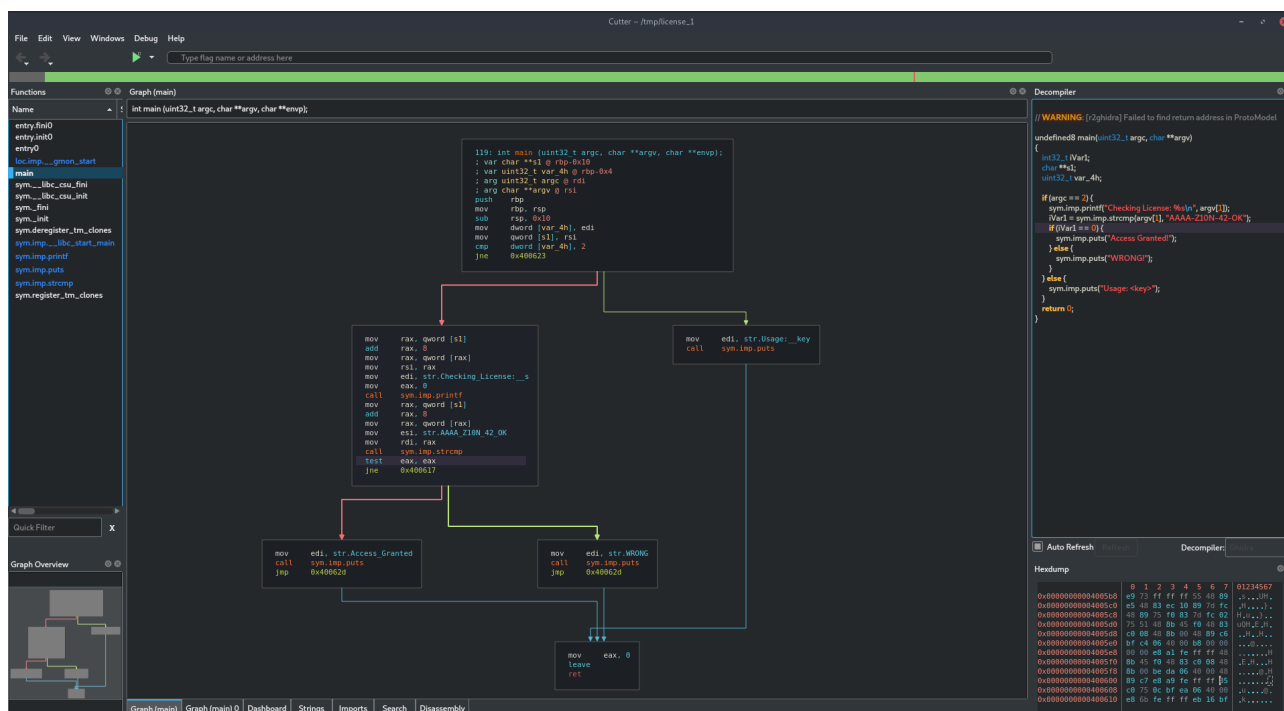


Illustration 2: izgled v brskalniku

Debugger napisan v jeziku Rust, ki uporablja knjižico ptrace za interakcijo s programom. Debugger bi omogočal nastaviti prekinitvene točke na določenih naslovih, lahko bi zbral vrednosti registra in jih spremenil, lahko bi izpisal zbirno kodo določene funkcije, tudi bi lahko prikazal vrednosti na stacku.

Potem bi naredili možnost pregleda v brskalniku z diagramom poteka (slika Illustration 2) in/ali polepšan konzolni pogled kot je prikazano na sliki Illustration 1.

Torej funkcije, ki jih bomo implementirali bodo:

- sledenje poteka izvajanja programa s prekinitvenimi točkami
- vpogled v registre in manipulacija z njimi
- vpogled v stack
- funkcija iskanja
- sledenje zgodovini vnesenih ukazov
- spremljanje celotnega izhoda
- pretvorba formata (hex v int v bin ipd.)
- lep pogled v ukaznoj vrstici z barvami
- GUI prikaz v spletnem brskalniku in komunikacija z programom v ozadju
- uporaba api za prenos informacij

Tehnologije za dejanski razvoj

Cargo (Rust package manager)

Vzpostavitev osnovnega časovnega načrta in natančnih opravil za naslednji teden

(Week 1) 8.11. - 14.11. (natancen plan v MS Planner-ju)

- create prompt (fdb) for input
- run / r run the program
- lf list all functions

(Week 2) 15. 11. - 21. 11.

- b / break [address] set breakpoint at given address
- del break [n] delete breakpoint number [n]
- d [func] disassemble function

(Week 3) 22. 11. - 28. 11.

- list break / lb list all breakpoints
- continue / c continue execution

(Week 4) 29. 11. - 5. 12.

- step / s step one instruction
- reg print values in all registers
- reg [name] print value in [name] register

(Week 5) 6. 12. - 12. 12

- set reg [name] [value] set register [name] to value [value]
- mem [address] [n] dump memory, [n] bytes starting from [address]
- stack dump memory from current stack frame

(Week 6) 13. 12. - 19. 12

- saving history of previously used commands
- converting formats (hex to int to bin, ...)

(Week 7) 20. 12. - 26. 12

- making output look nicer in console

(Week 8) 27. 12. - 2. 1.

- happy holidays

(Week 9) 3. 1. - 9. 1.

- implementing api calls for web GUI (for rust)
- implementing api calls for web GUI (for web)

(Week 10) 10. 1. - 16. 1.

- optimizing code, testing, making web GUI nicer

(Week 11) 17. 1. - 19. 1.

- optizmizing code, testing, (minor fixes)

Natančen plan za naslednji teden (8. 11. - 14. 11.)

RP

RPO Project ☆

...


Board

Charts

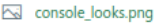
Schedule ...

Project Info

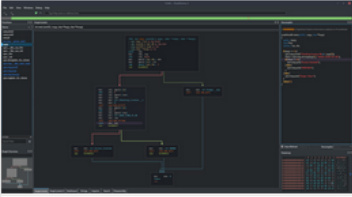
+ Add task




○ Console looks

 console_looks.png

🔗 1




○ Web looks

 web_look.png

🔗 1

○ Functions to implement

 functionality.txt

🔗 1

Ideas

+ Add task

To do

+ Add task

○ list functions

○ Print all function labels

○ Read bytes from functions

○ Setup instruction decoder for Rust (iced_x86)

🕒 0/3

IC Ivor Canjuga

○ run feature

○ Create struct that holds program data (PID, s

○ Implement run function that runs the progra

○ Print exit info (program exited successfully, S

📅 11/14 🕒 0/3

AB MZ

○ getting user input

○ Get user input (example: run / r) and call the

○ Add support for arguments ("break arg" call

📅 11/14 🕒 0/2

MK Marko Kričej

○ fdb input prompt

○ Create main loop

○ Add nice prompt

○ Create exit function that exits the debugger

📅 11/14 🕒 0/3

AB Alex Brence

Illustration 3: Natančen plan za 2. teden v nobembru