**МАТЕМАТИЧКА ГИМНАЗИЈА**

**У БЕОГРАДУ**

**МАТУРСКИ РАД**

**из предмета**

**Рачунарства и информатике**

на тему:

Основе оперативних система

Ментор

Никола Тасић

Ученик

Борис Ћеранић

**Београд, Мај 2023.**

# **Увод**

Технолошка развића донеле су многе промене у живот просечног човека, само дигитални уређаји трансформисали су начине на које комуницирамо, баратамо информацијама, складиштимо драгоцена сећања у виду слика и бивамо продуктивни у друштву. Још од почетка друге половине 20-ог века тече непрекидан развој рачунара, та категорија се касније шири на мобилне телефоне, конзоле за игрице, паметне фрижидере... Данас је опус разноврсности дигиталних уређаја готово непојмљив једном човеку, они се увлаче у сваки аспект људског живота. Свим тим уређајима је потребно нешто да их управља, неки софтвер, преферабилно што стандардизованији могући. У супротном би завладао хаос, за сваки уређај од њих бесконачно би било потребно писати различиту верзију софтвера. Тај проблем решавају, или барем ублажавају оперативни системи.

# Оперативни систем

Оперативни систем је софтвер или програм који контролише операције компјутерског система и његових ресурса. Осим тога, постоји једна веома важна критерија која је заједничка за све оперативни системе, а то је, да је оперативни систем способан да учитава и извршава програме независно од хардвера (уређаја) на коме се налазе док им омогућава стандардизован интерфејс за улаз и излаз. Неке од главних функција које опепративни систем може имати су:

* Управљање меморијом, уређајима за складиштење и другим системским ресурсима
* Одржавање реда и спровођење сигурносних полиса другим програмима
* Извршавање више програма истовремено, одређивање редоследа и важности извршавања,

давајући им одговарајућу предност – multitasking

* Динамичко покретање и гашење других програма
* Обезбеђење основног корисничког интерфејса као и *API*-ова

Не подржава сваки оперативни систем све ове функције. *MS-DOS*, на пример, није могао да да извршава више програма од једном, био је *single-tasking*. Важно је такође напоменути шта оперативни системи нису:

* Комјутерски хардвер
* Нека специфична апликација као што су игрице, процесори текста и *Web browser*-и.
* Колекција алатки попут *GNU*
* Окружење за развој софтвера, додуше поједији, попут *UCSD Pascal*, поседују интегрисана окужења
* Графички корисники интерфејс (GUI) без обзира на то што већина оперативних система долази са једним

# Историјат

Убаци историјат овде ово је лако са википедије...

# Методологија и опсег

У опсег овог рада спадају, поводом ограниченог времена, само неки аспекти оперативних система. Ради дубљег и лакшег разумевања рад ће бити обрађен из перспективе прављења једног врло елементарног оперативног система. То потенцијано укључује коришћење стандарда и пракси које су се користиле у прошлости, 90-их година прошлог века, а сада само заузимају место на старим интернет форумима или чак у музејима. Новији стандарди и праксе су много компликованији, али раде на сличном или истом принципу, чиме се тематика овог рада не нарушава.

## Boris-toolchain

## SmallerC

Smaller C је једноставан и мали single-pass C компајлер, тренутно подржава већину фукција C језика између *C89* и *C99* стандарда. *Self-host*-ован је на оперативним системима DOS, Windows и Linux, тиме може и да се покрене на њима и да компајлује програме за њих. Погодан је за писање оперативних система у различитим модовима процесора попут *Real Mode*, *Virtual 8086 Mode*, *Unreal Mode* и *32-bit Protected mode.*

Компајлер је праћен Preprocessor-ом и Linker-ом који може да произведе програмске фајл формате попут *COM* и *MZ* који користи *DOS, PE* који користи *Windows* и *ELF* који користи *Linux* оперативни систем. Подржава и стандардну C библиотеку написану за ове оперативне системе. За правилно функционисање му је потребан асемблер. Направљен је од стране *alexfru*-а и може се видети на GitHub-у путем [линка](https://github.com/alexfru/SmallerC).

Овај компајлер је поприлично згодан за писање једноставних оперативних система тиме што подржава *Unreal Mode* и поприлично је, по својој сврси, лаган и ненабуџен. Мане су додуше то што је врло слабо документиран као и недостатак људи који га користе чиме је подршка и помоћ страшно лимитирана.

## *NASM*

The Netwide Assembler- асемблер и дисасемблер намењен за Intel x86 архитектуру рачунара. Користи се за писање *16-Bit*, *32-Bit* и *64-Bit* програма. Оригинално написан од стране Симона Татхама. Подржава више излазних формата као *ELF* и .bin који се често користи за писање оперативних система.

Ми ћемо га већински користити индиректно преко *SmallerC*-а, а и помало директно где нам је потребан већи степен контроле поготово при првобитном покретању оперативног система. Директно написан assembly код биће примарно 16-Bit-ни.

## *Make*

Make је GNU алатка која је намењена за контролу генерације извршних фајлова и осталих пропратних фајлова приликом компилације и прављења програма. Може да се користи и за покретање направљених програма. Користи такозване *Makefile*-ове да би одредио редослед баратања фајлова као и који фајлови су потребни у ком тренутку.

Један *Makefile* садржи обично више правила које изгледају овако:

target: dependencies ...

commands

...

Где су *dependencies* target-и потребни да би се извршило правило, а commands команде које ће правило покренути.

# x86 Архитектура – кратак подсетник

## Регистри

## Stack

# **Делови оперативног система**

# MBR - boot

# Kernel

## Loader

## Менаџер меморије

## IPC – међипроцеска комуникација

# User-space

## Време

## File manager