

SVEUČILIŠTE JOSIPA JURAJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET OSIJEK

Patrik Pracny

**Implementacija sustava za pamćenje zadataka s
kružnim bufferom**

PROJEKTNI ZADATAK

30.1. Osijek, 2025.

1. UVOD

Projekt je izrađen kako bi omogućio efikasno upravljanje zadacima pomoću kružnog buffera. Rješava problem ograničenog kapaciteta memorije i omogućuje brzo dodavanje, uklanjanje i pregled zadataka bez potrebe za kompleksnim strukturama podataka.

Ciljevi zadatka:

- Implementirati sustav za pohranu i upravljanje zadacima.
- Koristiti kružni buffer kako bi se osigurala optimizacija memorijskog prostora
- Omogućiti validaciju unosa i osnovno rukovanje greškama.

U mnogim aplikacijama za upravljanje zadacima javlja se potreba za fiksnim brojem unosa koji se recikliraju nakon popunjavanja kapaciteta. Ovaj projekt omogućava efikasno rukovanje takvim podacima koristeći kružni buffer.

2. OPIS PROJEKTA

2.1. GLAVNE KOMPONENTE

- **Statička biblioteka** – Implementacija kružnog buffera.
- **Glavna aplikacija** – Interakcija s korisnikom i upravljanje zadacima.
- **Podatkovne strukture** – Struktura za spremanje zadataka i manipulacija njima.

2.2. FUNKCIONALNOSTI

Osnovne funkcionalnosti:

- Unos novog zadatka
- Označavanje zadatka kao završenog
- Ispis svih zadataka
- Ispis po prioritetu
- Izlaz iz programa
- Brisanje zadataka
- Brisanje cijelog buffera
- Spremanje zadataka u datoteku „tasks.txt“

2.3. STRUKTURA DIREKTORIJA

- kv_projekt/ (glavni direktorij)
 - src/ → izvorni kod aplikacije (sadrži .c datoteke)
 - include/ → Header datoteke za deklariranje funkcija i struktura
 - lib/ → Statička biblioteka s implementacijom kružnog buffera (sadrži .a datoteke u linux okruženju, dok u windows .lib)
 - obj/ → Objektne datoteke generirane tijekom kompilacije
 - bin/ → Izvršna datoteka aplikacije

3. INSTALACIJA I POKRETANJE

Da bi se preuzeo izvorni kod potrebno je u terminalu upisati slijedeće :

```
git clone https://gitlab.com/dock7713840/np\_kv\_projekt24.git
```

Nakon pokretanja naredbe, potrebno je ući u folder kv_projekt_pracny:

```
cd np_kv_projekt24
```

```
cd kv_projekt_pracny
```

Pošto ćemo kompilirati pomoću terminala potrebno je vidjeti da li imamo instaliran gcc na našem linux-u upisom slijedeće komande:

```
gcc --version
```

Komanda će nam ispisati verziju našeg gcc-a, u slučaju da nam to ne napravi potrebno je instalirati gcc.

Komanda za instalaciju gcc-a na Ubuntu distribuciji je:

```
apt install gcc
```

Nakon provjere (ili instalacije) potrebno je napraviti foldere koji će nam koristiti pri samoj kompilaciji i izvedbi aplikacije. Potrebno je napraviti slijedeće foldere „bin“, „lib“ i „obj“.

Foldere u terminalu pravimo pomoću komande:

```
mkdir naziv_foldere
```

Nakon što smo napravili potrebne foldere možemo krenuti na kompilaciju.

Prvo je potrebno kompilirati „object“ datoteke pomoću slijedećih naredbi:

```
gcc -O3 -Wall -I./include -c ./src/circular_buffer.c -o ./obj/circular_buffer.o
```

```
gcc -O3 -Wall -I./include -c ./src/task_manager.c -o ./obj/task_manager.o
```

```
gcc -O3 -Wall -I./include -c ./src/main.c -o ./obj/main.o
```

Nakon kompilacije potrebno je napraviti statičku biblioteku:

```
ar rcs ./lib/lib_project.a ./obj/circular_buffer.o ./obj/task_manager.o ./obj/main.o
```

Također, onda trebamo povezati „object“ datoteke s našom statičkom bibliotekom, te je potrebno napraviti izvršnu datoteku u našem slučaju nazvati ćemo je „Run“. To radimo preko slijedeće komande:

```
gcc -O3 -Wall -I./include -o ./bin/Run ./obj/main.o ./lib/lib_project.a
```

Kada smo to sve napravili aplikaciju možemo pokrenuti s slijedećom komandom:
`bin/./Run` ili `./Run` ako se nalazimo u folderu `bin`.

4. TEHNIČKI DETALJI

Unutar datoteka zaglavlja (`circular_buffer.h` i `task_manager.h`), deklarirane su sve funkcije i strukture podataka koje se koriste kroz ovaj projekt. Projekt se koristi kružnim bufferom kako bi se ostvarila organizacija podataka o zadacima. Unutar projekta nalaze se dvije glavne strukture, a to su:

- **Tasks:** sadrži sve podatke vezane za zadatke.
- **Circular Buffer:** koji sadrži strukturu **Tasks**, količinu elemenata i pointere na „head“ i „tail“.

Uz strukture, unutar datoteka nalaze se sljedeće glavne funkcije.

U `circular_buffer.h`:

- `void initialize(circular_buffer *cb);` → Inicijalizira strukturu
- `void addElement(circular_buffer *cb, Task *ts);` → Dodaje element u strukturu
- `int isEmpty(circular_buffer *cb);` → Provjerava da li je struktura prazna
- `int isFull(circular_buffer *cb);` → Provjerava da li je struktura puna
- `void printBuffer(circular_buffer *cb);` → Ispisuje buffer tj. strukturu
- `void clearBuffer(circular_buffer *cb);` → Čisti cijeli buffer

U `task_manager.h`:

- `void showMainMenu();` → Funkcija za glavni meni
- `void addTask();` → Funkcija za unos novog zadatka
- `void displayTasks(circular_buffer *cb);` → Funkcija za prikaz zadataka
- `void selectTask(circular_buffer *cb, int taskIndex, Task *task);` → Funkcija za odabir zadatka
- `void markTaskCompleted(Task *task);` → Funkcija za označavanje zadatka kao završenog
- `void deleteTask(circular_buffer *cb, int taskIndex);` → Funkcija za brisanje zadatka

Kružni buffer je jedan od bitnijih algoritama jer omogućuje efikasnu strukturu podataka koja koristi pokazivače za optimizaciju dodavanje i uklanjanja elemenata, te isto bitan algoritam je sortiranje po prioritetu koje omogućuje ispis zadataka prema važnosti.

5. ZAKLJUČAK

Projekt "Task Manager" omogućuje efikasno upravljanje zadacima koristeći kružni buffer. Implementacija osigurava optimizirano korištenje memorije, jednostavnost korištenja i modularnost koda.

Moguća buduća poboljšanja:

- Dodavanje grafičkog sučelja za korisnike.
- Dodavanje algoritama pretraživanja i filtriranja zadataka.

6. PRILOZI

U nastavku su priloženi primjeri toka testiranja i pokretanja programa.

```
cobalt@Cobalt11:~/Docs/np_kv_projekt24/kv_projekt_pracny$ bin/./Run
No tasks file found, starting with an empty buffer.

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: 1
Unesite naslov zadatka (maks. 50 znakova): Predaj projektni zadatak.
Odaberite prioritet (1 - LOW, 2 - MEDIUM, 3 - HIGH): 3
Unesite dan: 31
Unesite mjesec: 1
Odaberite status (1 - ACTIVE, 2 - FINISHED): 1
Zadatak uspješno dodan.

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: |
```

Pokretanje aplikacije i unos zadatka.


```
===== Zadaci =====

===== Detalji zadatka =====
Naslov: Predaj projektni zadatak.
Prioritet: 3
Datum: 31/01
Status: ACTIVE
=====

===== Detalji zadatka =====
Naslov: Provjera prioriteta1
Prioritet: 2
Datum: 10/01
Status: ACTIVE
=====

===== Detalji zadatka =====
Naslov: Provjera prioriteta
Prioritet: 1
Datum: 10/01
Status: ACTIVE
=====
=====
```

Nakon unosa zadataka, prikaz zadataka sortiranih po prioritetu.

```

Odaberite broj zadatka (0 za povratak): 1

===== Detalji zadatka =====
Naslov: Predaj projektni zadatak.
Prioritet: 3
Datum: 31/01
Status: ACTIVE
=====
1. Označi kao dovršen
2. Obriši zadatak
3. Povratak
Odabir: 2
Zadatak na indeksu 1 je obrisao.
Zadatak uspješno obrisao
Odaberite broj zadatka (0 za povratak): 0
Povratak u glavni meni.

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: 2
Zadaci su uspješno sortirani prema prioritetu.

===== Zadaci =====

===== Detalji zadatka =====
Naslov: Predaj projektni zadatak.
Prioritet: 3
Datum: 31/01
Status: ACTIVE
=====

===== Detalji zadatka =====
Naslov: Predaj projektni zadatak.
Prioritet: 3
Datum: 31/01
Status: ACTIVE
=====
=====
Odaberite broj zadatka (0 za povratak):

```

Nakon prikaza zadatka odabiremo zadatak s indexom 2 i brišemo ga.

```

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: 3
Tasks saved to file successfully.
Izlazak iz programa...
cobalt@Cobalt11:~/Docs/np_kv_projekt24/kv_projekt_pracny$ ls -al
total 36
drwxr-xr-x 8 cobalt cobalt 4096 Jan 31 12:34 .
drwxr-xr-x 4 cobalt cobalt 4096 Jan 31 12:13 ..
drwxr-xr-x 3 cobalt cobalt 4096 Jan 31 11:38 Debug
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:41 bin
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:38 include
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:41 lib
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:40 obj
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:38 src
-rw-r--r-- 1 cobalt cobalt  70 Jan 31 12:34 tasks.txt
cobalt@Cobalt11:~/Docs/np_kv_projekt24/kv_projekt_pracny$ |

```

Spremanje izmjena u datoteku „tasks.txt“

```

Tasks loaded from file successfully.

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: 4
Da li sigurno hoćeš obrisati sve? Upiši da/ne: da
Tasks deleted successfully.
Buffer cleared and tasks deleted.
Buffer je izpražnjen

===== Glavni Meni =====
1. Dodaj novi zadatak
2. Prikaži zadatke
3. Spremi u tasks.txt
4. Očisti buffer.
5. Izlaz
=====
Odabir: 5
Izlaz iz programa.
cobalt@Cobalt11:~/Docs/np_kv_projekt24/kv_projekt_pracny$ ls -al
total 32
drwxr-xr-x 8 cobalt cobalt 4096 Jan 31 12:41 .
drwxr-xr-x 4 cobalt cobalt 4096 Jan 31 12:13 ..
drwxr-xr-x 3 cobalt cobalt 4096 Jan 31 11:38 Debug
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:41 bin
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:38 include
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:41 lib
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:40 obj
drwxr-xr-x 2 cobalt cobalt 4096 Jan 31 11:38 src
cobalt@Cobalt11:~/Docs/np_kv_projekt24/kv_projekt_pracny$ |

```

Brisanje cijelog buffera i datoteke.

