



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КІЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп'ютерних
систем**

Лабораторна робота №3

З дисципліни «Мережеві операційні системи»

На тему: «Керування пам'яттю»

Виконав: студент IV курсу

Групи КВ-73

Тарновський А.М.

Технічне завдання

14. *Кеш пам'ять.* Розробити модель, яка реалізує алгоритм пошуку та заміщення даних в кеш пам'яті для випадкового відображення даних на кеш. Узгодження даних основної та кеш пам'яті виконати за принципом зворотного запису

При моделюванні алгоритмів роботи кеш пам'яті, адресний простір кеш пам'яті і основної пам'яті може задаватися довільно при співвідношенні їх обсягів не менш ніж 1:10. Пошук, запис і заміщення інформації в кеш пам'яті повинно виконуватися шляхом завдання шуканих адрес основної пам'яті.

Лабораторна робота була виконана у вигляді сайту, стилізованого під псевдо-графічний інтерфейс MS-DOS (рис 1.1). Сторінка сайту поділяється на три блоки:

- Термінал для управління контроллером пам'яті
- Таблиця даних оперативної пам'яті
- Таблиця даних КЕШ-пам'яті

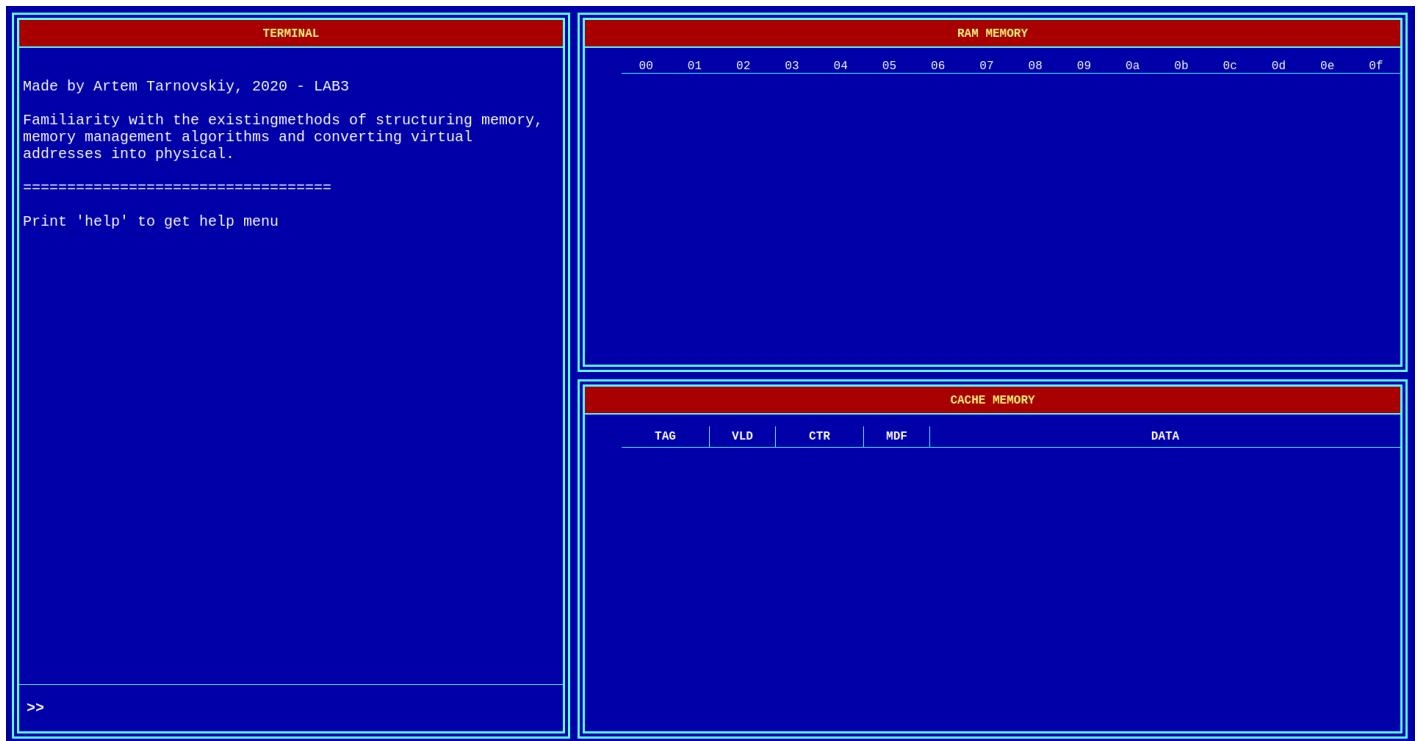


Рисунок 1.1 — загальний вигляд програми

Дані оперативної пам'яті відображаються у вигляді сторінок розміром 16 елементів, кожен з яких має розмір 2 байти. Дані КЕШ-пам'яті відображаються у вигляді КЕШ-ліній, кожна з яких має розмір 64 байти:

- 2 байти адреси
- 2 байти керуючої інформації (1 байт лічильник запитів, 4 біти — ознака модифікації даних та 4 біти — ознака актуальності даних)
- 60 байт даних

Такі розміри були вибрані для зручності демонстрації роботи моделі КЕШ-пам'яті.

Демонстрація роботи

Для початку роботи необхідно задати розміри КЕШ та RAM пам'яті. Це можна зробити за допомогою команди set (рис 2.1). За варіантом завдання відношення об'єму ОП до КП має бути більшим за 10, тому для демонстрації роботи візьмемо наступні значення: для ОП — 16Кб, КП — 256 байт. Ці розміри далекі від сучасних, але для принципу роботи це не важливо.

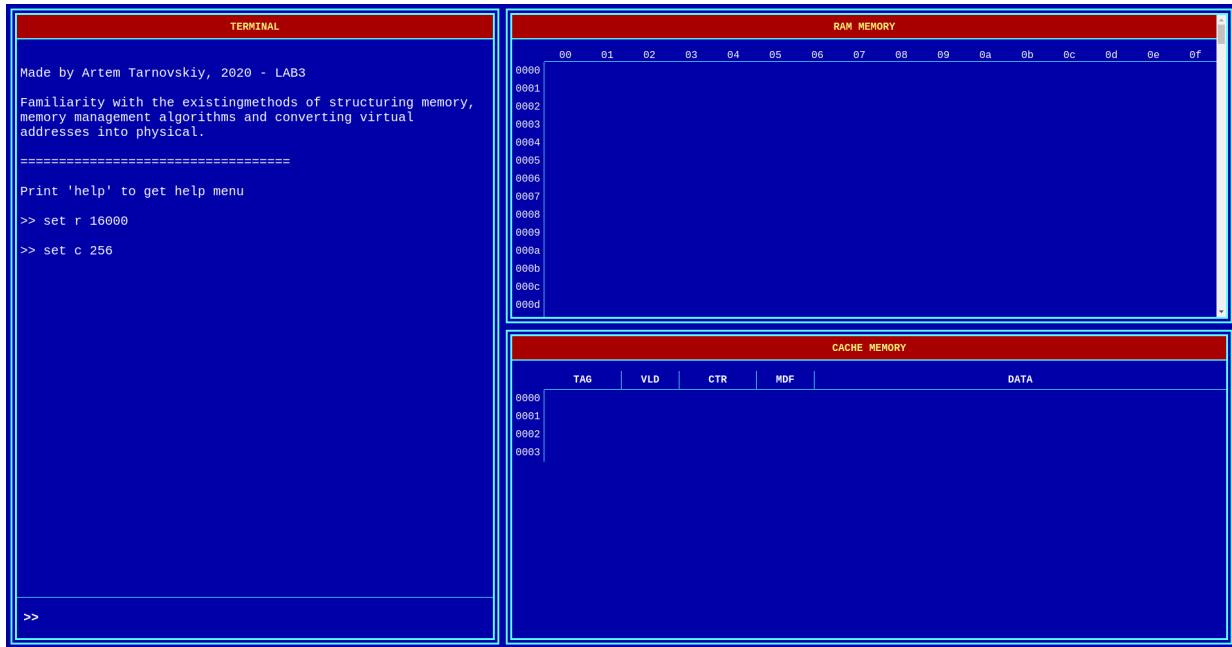


Рисунок 2.1 — вигляд програми після задання розмірів

Щоб згенерувати данні для подальшої роботи виконаємо команду gen. Ця команда генерує по 2 байти інформації для кожної комірки оперативної пам'яті (рис 2.2).

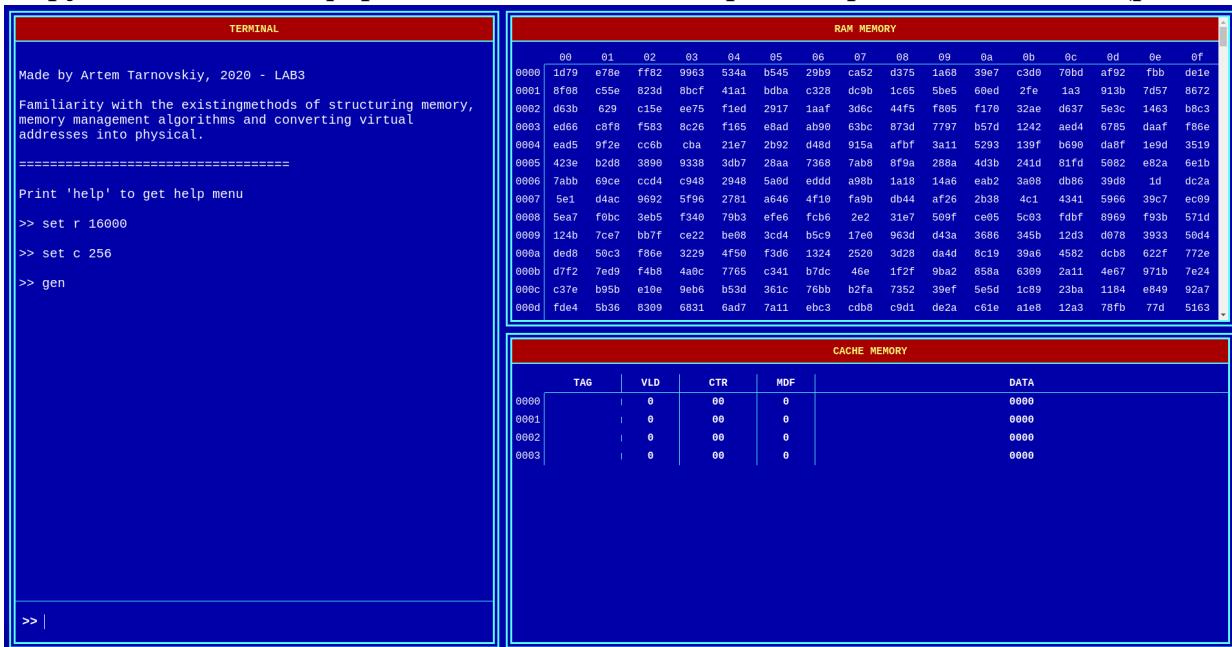


Рисунок 2.2 — вигляд програми після генерації даних

При генерації даних для ОП генерується потрібна к-сть КЕШ-ліній. За замовчуванням у КП дані відсутні.

Оскільки користувач виступає в ролі «процесора», який дає команди контроллеру пам'яті, то для керування даними використовується термінал. Для отримання даних необхідно виконати команду get <address> (рис 2.3)

RAM MEMORY																
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb	de1e
0001	8f08	c55e	823d	8bcf	41a1	bdba	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	d63b	629	c15e	e75	f1ed	2917	1aa	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	a040	63bc	873d	7797	b57d	1242	aea4	6785	daaf	f86e
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afb	3a11	5293	139f	b690	da8f	1e9d	3519
0005	423e	b2d8	3890	9338	3db7	28aa	7368	7ab8	8f9a	288a	4d3b	241d	81fd	5082	e82a	6e1b
0006	7abb	69ce	cccd	c948	2948	5a0d	eddd	a98b	1a10	14a6	eab2	3a08	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	a7e2	2b38	4c1	4341	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	ef6	fcb6	2e2	31e7	509f	ce05	5c03	fd8	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d678	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dcb8	622f	772e
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cdb8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY															
	TAG	VLD	CTR	MDF	DATA										
0000	8a	1	00	0	ce05										
0001		0	00	0	0000										
0002		0	00	0	0000										
0003		0	00	0	0000										

Рисунок 2.3 — отримання даних

При виконанні команди get контролер спочатку виконує пошук у КП, якщо елемент не знайдено, то контроллер звертається до ОП, молодший із виділених 4-х біт) у лінії КП відповідає за актуальність даних. Якщо VLD == 0, то лінія позначається як не потрібна та може бути перезаписана. CTR — лічильник запитів за тегом (адресою) до даних (відображається у програмі 2 молодших біти із 8-х виділених). Це поле необхідно для контролера пам'яті, щоб у разі витіснення данні для витіснення вибиралися за найменшою к-стю звертань. MDF — ознака модифікації даних (відображається молодший з 4-х виділених біт).

Виконаємо ще кілька запитів за іншими адресами пока не заповнимо кеш-пам'ять, та кілька запитів за адресами що є у КП (рис 2.4)

TERMINAL

```

CACHE memory...
e78e
>> get ae
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
622f
>> get 3
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
9963
>> get ae
622f
>> get ae
622f
>> get ae
622f
>> get 8a
ce05
>> get 1
e78e
>> get 1
e78e
>>

```

RAM MEMORY

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb	de1e
0001	8f08	c55e	823d	8bcf	41a1	bdba	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	d63b	e29	c15e	ee75	f1ed	2917	1aa9	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf	f8e6
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b698	da8f	1e9d	3519
0005	423e	b2d8	3898	9338	3db7	28aa	7368	7ab8	8f9a	28a8	4d3b	241d	81fd	5682	e82a	6e1b
0006	7abb	69ce	ccdd	c948	2948	5a9d	eddd	a9b8	1a18	14a6	eab2	3a08	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	ef66	fcb6	2e2	31e7	509f	ce05	5c03	fd8f	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3886	345b	12d3	d078	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dc8b	622f	772e
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cdb8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY

TAG	VLD	CTR	MDE	DATA
0000	8a	1	1	0
0001	1	1	2	0
0002	ae	1	3	0
0003	3	1	00	0

Рисунок 2.4 — вигляд програми при заповненні кеш пам’яті та неодноразових запитів до даних у КП

Після кількох запитів контролер помістив у КП дані за адресами 0001, 00AE, 0003 та зібльшив лічильник у залежності від к-сті запитів до кожної з адрес (окрім 0003, оскільки за цією адресою ми зверталися лише один раз).

Спробуємо змінити дані. Для цього необіхдно виконати команду update (рис 2.5).

TERMINAL

```

e78e
>> get ae
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
622f
>> get 3
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
9963
>> get ae
622f
>> get ae
622f
>> get ae
622f
>> get 8a
ce05
>> get 1
e78e
>> get 1
e78e
>> update ae FAFA
>>

```

RAM MEMORY

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb	de1e
0001	8f08	c55e	823d	8bcf	41a1	bdba	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	d63b	e29	c15e	ee75	f1ed	2917	1aa9	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf	f8e6
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b698	da8f	1e9d	3519
0005	423e	b2d8	3898	9338	3db7	28aa	7368	7ab8	8f9a	28a8	4d3b	241d	81fd	5682	e82a	6e1b
0006	7abb	69ce	ccdd	c948	2948	5a9d	eddd	a9b8	1a18	14a6	eab2	3a08	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	ef66	fcb6	2e2	31e7	509f	ce05	5c03	fd8f	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3886	345b	12d3	d078	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dc8b	622f	772e
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cdb8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY

TAG	VLD	CTR	MDE	DATA
0000	8a	1	1	0
0001	1	1	2	0
0002	ae	1	3	0
0003	3	1	00	0

Рисунок 2.5 — вигляд програми після запиту на зміну даних

У нашому випадку контроллер працює за алгоритмом зворотнього запису. Як видно з рис. 2.5, при виконанні команди update ae FAFA дані в ОП залишились без змін (622f), але в КП дані змінились, а поле MDF змінює значення на 1 — це означає, що дані за цією адресою були змінені та при витісненні вони будуть занесені до ОП. Переконатись можна виконавши команду free (рис 2.6). Вона змусить виконати витіснення даних не зважаючи на алгоритм (витіснення відбувається лише при заповненні КП) роботи контролера. Така команда була введена у якості допоміжної для демонстрації роботи окремих блоків алгоритму.

The screenshot shows two windows side-by-side. The left window is titled 'TERMINAL' and contains a command-line session:

```

>> get ae
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
622f

>> get 3
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...
9963

>> get ae
622f

>> get ae
622f

>> get ae
622f

>> get 8a
ce05

>> get 1
e78e

>> get 1
e78e

>> update ae FAFA
>> free ae
>>

```

The right window is titled 'RAM MEMORY' and displays a memory dump with columns for addresses (00 to ff) and data bytes (0e to 0f). A green arrow points from the 'ae' entry in the terminal to the 'ae' row in the RAM dump.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb	de1e
0001	8f08	c55e	823d	8bcf	41a1	bdb4	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	d63b	629	c15e	ee75	fied	2917	1aaaf	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf	f86e
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afb4	3a11	5293	139f	b699	da8f	1e9d	3519
0005	423e	b2d8	3890	9338	3db7	28aa	7368	7ab8	8f9a	288a	43d3	411d	81fd	5682	e82a	6e1b
0006	7abb	69ce	cc4d	c948	2948	5a0d	eddd	a98b	1a18	14a6	eab2	3a8b	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4541	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	eфе6	fcb6	2e2	31e7	509f	ce05	5c03	fdfb	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	he08	3cd4	h5c9	1760	983d	d43a	3686	345b	12d3	d678	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dc8b	FAFA	772e
000b	d7f2	7ed9	f4b8	4a8c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cd8b	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY					
TAG	VLD	CTR	MDF	DATA	
0000	8a	1	1	0	ce05
0001	1	1	2	0	e78e
0002	ae	0	3	1	FAFA
0003	3	1	00	0	9963

Рисунок 2.6 — зображення програми після витіснення даних

Як видно з рис. 2.6 при витісненні даних поле актуальності (VLD) змінюється на 0, що означає, що при необхідності наступні дані можуть бути поміщені у цей рядок КП (рис 2.7). Оскільки ознака модифікації була рівна 1, то після витіснення з КП даних вони були оновлені у ОП за відповідною адресою.

TERMINAL

```

>> get 3
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...

9963

>> get ae
622f

>> get ae
622f

>> get ae
622f

>> get 8a
ce05

>> get 1
e78e

>> get 1
e78e

>> update ae FAFA

>> free ae

>> get 7b
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...

4c1

>> |

```

RAM MEMORY

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	7b9d	af92	fb
0001	8f08	c55e	823d	8bcf	41a1	bd8a	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57
0002	d63b	629	c15e	ee75	f1ed	2917	1aa9	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b699	da8f	1e9d
0005	423e	b2d8	3890	9338	3db7	28aa	7368	7ab8	8f9a	288a	4d3b	241d	81fd	5082	e82a
0006	7abb	69ce	cc4d	c948	2948	5a0d	eddd	a98b	1a18	14a6	eab2	3a08	db86	39d8	1d
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7
0008	5ea7	f0b8	3eb5	f340	79b3	ef66	fcb6	2e2	31e7	509f	ce05	5c03	fdbf	8969	f93b
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d078	3933
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dcb8	FAFA
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cd8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	7b9d	af92	fb
0001	8f08	c55e	823d	8bcf	41a1	bd8a	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57
0002	d63b	629	c15e	ee75	f1ed	2917	1aa9	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b699	da8f	1e9d
0005	423e	b2d8	3890	9338	3db7	28aa	7368	7ab8	8f9a	288a	4d3b	241d	81fd	5082	e82a
0006	7abb	69ce	cc4d	c948	2948	5a0d	eddd	a98b	1a18	14a6	eab2	3a08	db86	39d8	1d
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7
0008	5ea7	f0b8	3eb5	f340	79b3	ef66	fcb6	2e2	31e7	509f	ce05	5c03	fdbf	8969	f93b
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d078	3933
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dcb8	FAFA
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cd8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d

CACHE MEMORY

TAG	VLD	CTR	MDF	DATA
0000	8a	1	1	0
0001	1	1	2	0
0002	7b	1	00	0
0003	3	1	00	0

Рисунок 2.7 — вигляд програми після чергового запиту на отримання даних.

Як видно з рис. 2.7, при запиті за адресою 7B контролер помістив інформацію у вільний рядок КП (VLD == 0). Але основний алгоритм такий, що дані витісняються лише тоді, коли у КП не залишилось вільного місця. Маємо наступну картину: за адресою 8A — 2 запити (нумерація з 0), 0001 — 3 запити, 7B та 0003 — по одному. Отже при новому запиті на отримання даних контроллер пам'яті вибере одну з двох ліній, у якої тег 7B або 0003. Переконаємося у цьому (рис 2.8)

TERMINAL

```

>> get ae
622f

>> get ae
622f

>> get ae
622f

>> get 8a
ce05

>> get 1
e78e

>> get 1
e78e

>> update ae FAFA

>> free ae

>> get 7b
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...

4c1

>> get 0000
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...

4c1

>> get 0000
No element in cache found. Looking in RAM MEMORY...
Element was found in RAM. Getting element and pushing to CACHE memory...

4c1

>> |

```

RAM MEMORY

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	7b9d	af92	fb
0001	8f08	c55e	823d	8bcf	41a1	bd8a	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57
0002	d63b	629	c15e	ee75	f1ed	2917	1aa9	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b699	da8f	1e9d
0005	423e	b2d8	3890	9338	3db7	28aa	7368	7ab8	8f9a	288a	4d3b	241d	81fd	5082	e82a
0006	7abb	69ce	cc4d	c948	2948	5a0d	eddd	a98b	1a18	14a6	eab2	3a08	db86	39d8	1d
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7
0008	5ea7	f0b8	3eb5	f340	79b3	ef66	fcb6	2e2	31e7	509f	ce05	5c03	fdbf	8969	f93b
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d078	3933
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dcb8	FAFA
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cd8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d

CACHE MEMORY

TAG	VLD	CTR	MDF	DATA
0000	8a	1	1	0
0001	1	1	2	0
0002	0000	1	00	0
0003	3	1	00	0

Рисунок 2.8 — вигляд програми після витіснення даних за кількістю запитів

Для модифікації даних не обов'язково щоб вони були у КП. Щоб виконати модифікацію даних, які не містяться у КП необхідно виконати команду `put`. Команди `put` та `update` були розділені лише для зручності демонстрації роботи. Виконаємо очистку термінала командою `cls` та модифікацію даних на 0000 за адресою 0011 (рис 2.9). При виконані команди `put` дані заносяться до КП без втручання в ОП, відразу з $VLD = 1$ та $MDF = 1$. При витіснені дані будуть перезаписані у відповідну комірку ОП (Для впевненості, що витіснення буде виконуватись за потрібним нам тегом, збільшемо лічильник останньої КЕШ-лінії шляхом запиту даних за відповідною адресою) (рис 2.10).

The screenshot shows the system's internal state. On the left, the **TERMINAL** window displays the command `>> put 0011 0000`. On the right, the **RAM MEMORY** window shows a list of memory locations from 0000 to 000d. The entry at address 0011 has been updated to 0000, indicated by a red arrow. The **CACHE MEMORY** window below it shows a 4x4 grid. The entry at index 0002 (tag 0011) now contains 0000, also indicated by a red arrow.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb8	de1e
0001	8f08	c55e	823d	8bcf	41a1	bd8a	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	6d30	629	c15e	ee75	f1ed	2917	1aa1	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf	f86e
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b690	da8f	1e9d	3519
0005	423e	b2d8	3898	9338	3db7	28a8	7368	7ab8	879a	288a	4d3b	241d	81fd	5082	e82a	6e1b
0006	7abb	69ce	cccd4	c948	2948	5a0d	eddd	a98b	1a18	1a46	eab2	3a08	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	ef6e	fcb6	2e2	31e7	509f	ce05	5c03	fdbf	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d078	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dc8b	FAFA	772e
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cdb8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY				DATA
TAG	VLD	CTR	MDF	
0000	8a	1	1	ce95
0001	1	1	2	e78e
0002	0011	1	00	0000
0003	3	1	00	9963

Рисунок 2.9 — вигляд програми після запиту на зміну даних, яких ще немає в кеші.

The screenshot shows the system's internal state. On the left, the **TERMINAL** window displays the command `>> get 3 9963`, followed by the message "No element in cache found. Looking in RAM MEMORY...". It then shows "Element was found in RAM. Getting element and pushing to CACHE memory..." and the value `8c26`. On the right, the **RAM MEMORY** window shows the same data as in Figure 2.9. The **CACHE MEMORY** window below it shows a 4x4 grid. The entry at index 0002 (tag 0011) has been replaced by the value 8c26, indicated by a red arrow.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	1d79	e78e	ff82	9963	534a	b545	29b9	ca52	d375	1a68	39e7	c3d0	70bd	af92	fb8	de1e
0001	8f08	0000	823d	8bcf	41a1	bd8a	c328	dc9b	1c65	5be5	60ed	2fe	1a3	913b	7d57	8672
0002	6d30	629	c15e	ee75	f1ed	2917	1aa1	3d6c	44f5	f805	f170	32ae	d637	5e3c	1463	b8c3
0003	ed66	c8f8	f583	8c26	f165	e8ad	ab90	63bc	873d	7797	b57d	1242	aed4	6785	daaf	f86e
0004	ead5	9f2e	cc6b	cba	21e7	2b92	d48d	915a	afbf	3a11	5293	139f	b690	da8f	1e9d	3519
0005	423e	b2d8	3898	9338	3db7	28a8	7368	7ab8	879a	288a	4d3b	241d	81fd	5082	e82a	6e1b
0006	7abb	69ce	cccd4	c948	2948	5a0d	eddd	a98b	1a18	1a46	eab2	3a08	db86	39d8	1d	dc2a
0007	5e1	d4ac	9692	5f96	2781	a646	4f10	fa9b	db44	af26	2b38	4c1	4341	5966	39c7	ec09
0008	5ea7	f0bc	3eb5	f340	79b3	ef6e	fcb6	2e2	31e7	509f	ce05	5c03	fdbf	8969	f93b	571d
0009	124b	7ce7	bb7f	ce22	be08	3cd4	b5c9	17e0	963d	d43a	3686	345b	12d3	d078	3933	50d4
000a	ded8	50c3	f86e	3229	4f50	f3d6	1324	2520	3d28	da4d	8c19	39a6	4582	dc8b	FAFA	772e
000b	d7f2	7ed9	f4b8	4a0c	7765	c341	b7dc	46e	1f2f	9ba2	858a	6309	2a11	4e67	971b	7e24
000c	c37e	b95b	e10e	9eb6	b53d	361c	76bb	b2fa	7352	39ef	5e5d	1c89	23ba	1184	e849	92a7
000d	fde4	5b36	8309	6831	6ad7	7a11	ebc3	cdb8	c9d1	de2a	c61e	a1e8	12a3	78fb	77d	5163

CACHE MEMORY				DATA
TAG	VLD	CTR	MDF	
0000	8a	1	1	ce95
0001	1	1	2	e78e
0002	33	1	00	0
0003	3	1	1	9963

Рисунок 2.10 — витіснення модифікованих даних яких не було в кеші до модифікації