**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №9

по дисциплине: **«**[**Операционные системы**](https://www.edu.gstu.by/course/view.php?id=557)**»**

**Алгоритмы замещения страниц**

Выполнил: студент гр. ИТП-11

Горкунов А.О,

Принял: Преподаватель

Карась О.В.

Гомель 2023

**Цель работы**: изучение алгоритмов управления памятью, разработка программы менеджера памяти*.*

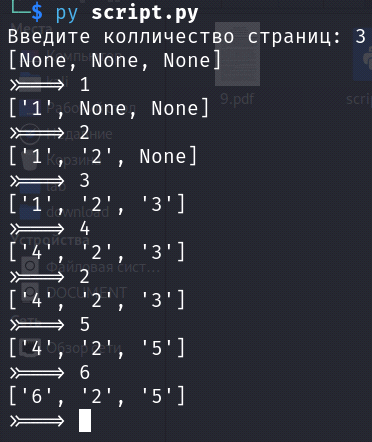
**Ход работы**

***Задание.* Разработать программу, реализующую заданный алгоритм замещения страниц в памяти**

**1. Создание файла script.py реализующего алгоритм замещения страниц в памяти**

**2. Тестирование работы скрипта**

На рисунке 1 представлено создание памяти размером 3 страницы, и заполнение её страницами



*Рисунок 1 - создание памяти и заполнение её страницами*

**Вывод: мы разработали программу, реализующую заданный алгоритм замещения страниц в памяти.**

Листинг:

class LruClock:

def \_\_init\_\_(self, page\_count):

self.page\_count = page\_count

self.pages = []

self.counter = 0

for \_ in range(page\_count):

self.pages.append({'page\_name': None, 'count\_register': 0})

def \_\_str\_\_(self):

return str([page['page\_name'] for page in self.pages])

def access\_page(self, page\_name):

self.counter += 1

for index, page in enumerate(self.pages):

if page['page\_name'] == page\_name:

self.pages[index]['count\_register'] = self.counter

break

else:

min\_counter = self.counter

imin\_counter = 0

for index, page in enumerate(self.pages):

if page['count\_register'] < min\_counter:

min\_counter = page['count\_register']

imin\_counter = index

self.pages[imin\_counter] = {'page\_name': page\_name, 'count\_register': self.counter}

def main() -> None:

pages\_count = int(input("Введите колличество страниц: "))

lru\_clock = LruClock(pages\_count)

while True:

print(lru\_clock)

number\_pages = input(">>==> ")

lru\_clock.access\_page(number\_pages)

if \_\_name\_\_ == "\_\_main\_\_":

try:

main()

except KeyboardInterrupt:

print("\nExit.")