



Faculty of Technology and Engineering

Chandubhai S. Patel Institute of Technology

Department of Computer Science & Engineering

Date: 22/03/25

Practical 8

Academic Year	:	2023-24	Semester	:	4 th
Course code	:	CSE208	Course name	:	Operating System

Perform Linux Commands for the following

Practical 8: Simulating Page Replacement Algorithms

You are a memory management engineer tasked with optimizing the performance of a virtual memory system. Your goal is to simulate and evaluate the behavior of two page replacement algorithms:

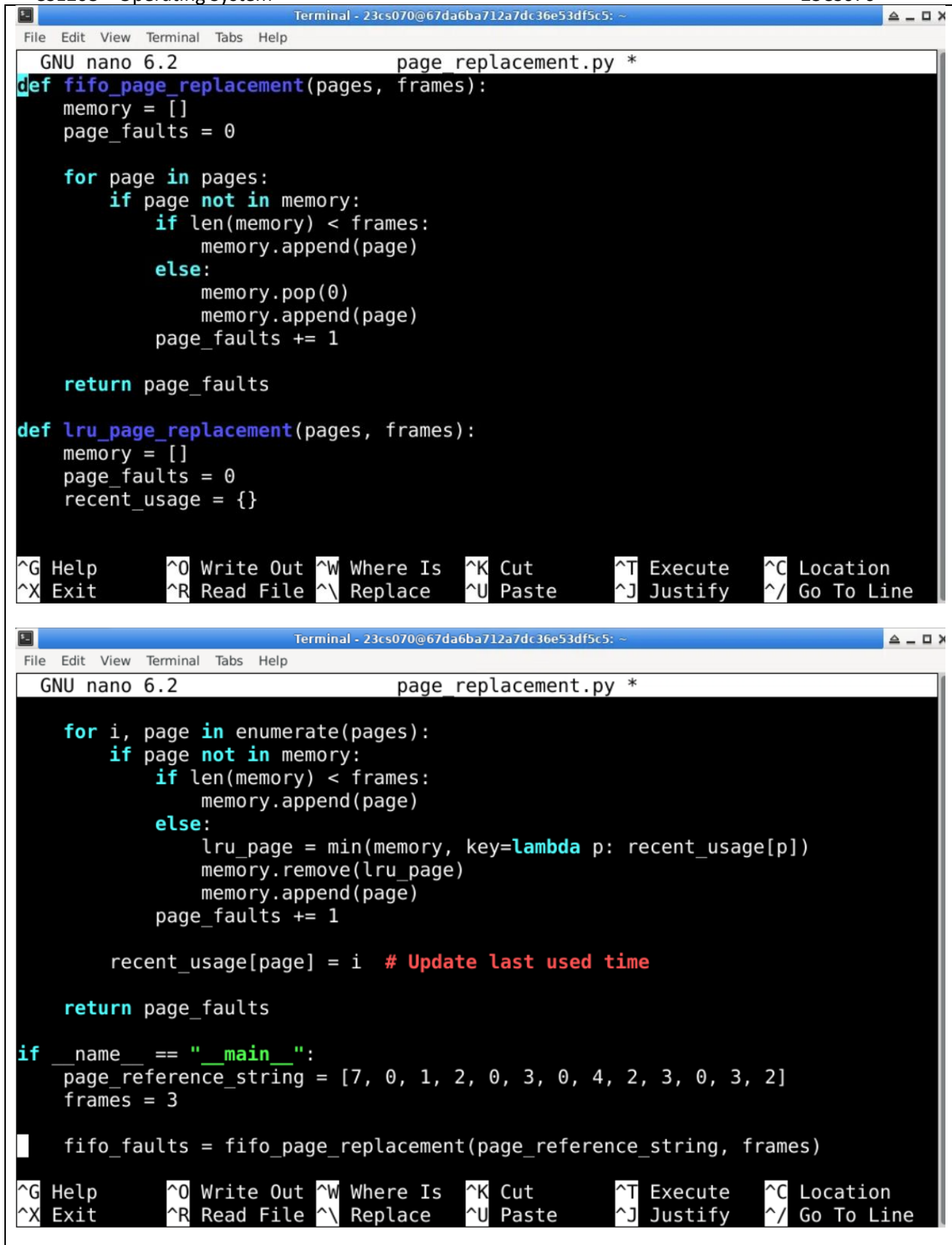
1. **FIFO (First-In-First-Out)**
2. **LRU (Least Recently Used)**

Project Scenario:

A system has **3 page frames** available for storing pages in memory. A sequence of page requests arrives, and your task is to determine how efficiently the algorithms handle page faults.

- **Page Reference String:** 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2
- **Objective:** Minimize the number of page faults.

```
23cs070@67da6ba712a7dc36e53df5c5:~$ nano page_replacement.py
```



The image shows two screenshots of a terminal window with the nano 6.2 editor. The terminal title bar indicates the user is 23cs070@67da6ba712a7dc36e53df5c5. The file being edited is page_replacement.py.

The first screenshot shows the initial code for the FIFO page replacement algorithm:

```
GNU nano 6.2 page_replacement.py *
def fifo_page_replacement(pages, frames):
    memory = []
    page_faults = 0

    for page in pages:
        if page not in memory:
            if len(memory) < frames:
                memory.append(page)
            else:
                memory.pop(0)
                memory.append(page)
            page_faults += 1

    return page_faults

def lru_page_replacement(pages, frames):
    memory = []
    page_faults = 0
    recent_usage = {}
```

The second screenshot shows the code after adding the LRU page replacement algorithm and a main function to test it:

```
GNU nano 6.2 page_replacement.py *

    for i, page in enumerate(pages):
        if page not in memory:
            if len(memory) < frames:
                memory.append(page)
            else:
                lru_page = min(memory, key=lambda p: recent_usage[p])
                memory.remove(lru_page)
                memory.append(page)
            page_faults += 1

        recent_usage[page] = i # Update last used time

    return page_faults

if __name__ == "__main__":
    page_reference_string = [7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2]
    frames = 3

    fifo_faults = fifo_page_replacement(page_reference_string, frames)
```

```
Terminal - 23cs070@67da6ba712a7dc36e53df5c5: ~
File Edit View Terminal Tabs Help
GNU nano 6.2 page_replacement.py *
    memory.append(page)
    page_faults += 1

    recent_usage[page] = i # Update last used time

    return page_faults

if __name__ == "__main__":
    page_reference_string = [7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2]
    frames = 3

    fifo_faults = fifo_page_replacement(page_reference_string, frames)
    lru_faults = lru_page_replacement(page_reference_string, frames)

    print("\nFIFO Page Replacement:")
    print(f"Total Page Faults: {fifo_faults}")

    print("\nLRU Page Replacement:")
    print(f"Total Page Faults: {lru_faults}")

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line

23cs070@67da6ba712a7dc36e53df5c5:~$ python3 page_replacement.py

FIFO Page Replacement:
Total Page Faults: 10

LRU Page Replacement:
Total Page Faults: 9
```