

# Experiment No. 7

Title: LRU page replacement algorithm

Batch: B-2 Roll No: 16010422234 Name: Chandana Galgali Date: 04/10/2024

# Experiment No: 7

\_\_\_\_\_

Aim: To implement the Least Recently Used (LRU) page replacement algorithm.

\_\_\_\_\_

Resources needed: Text editor and JAVA/C compiler.

\_\_\_\_\_

#### Theory:

## Pre lab/Prior concepts:

Algorithm:

- 1. Accept the number of frames from the user in n.
- Accept the size of the reference string in m.
- Read reference string into array s.
- Take a stack of maximum size=n.
- Read one page number from s and enter it in the stack. Increment stack pointer.
- Read the second page number from s. compares it with the page number existing in the stack. If matches then hit++ and goto step 7 else goto step 8.
- Remove that page from the stack. Shift all the contents of the stack down by one.
   And put the current page on stack top. (least recently used page will be at the bottom of stack)
- If the stack is empty (frame is free) then insert the current page on the stack top
  else shift the content of the stack down by one and insert the current page on the
  stack top.
- Print the current stack.
- Repeat steps 6 to 9 till all the content of reference string is processed
- Calculate and print Hit ratio.
- 12. Stop.

\_\_\_\_\_

#### Results:

```
import matplotlib.pyplot as plt
import pandas as pd

def lru_page_replacement(n, reference_string):
    stack = [] # Initialize an empty stack
    hits = 0
    misses = 0
    frame_states = [] # To store the state of frames after each page
```

```
request
   results = [] # To store hit/miss results
  for page in reference string:
      if page in stack:
          # Page hit
          hits += 1
          stack.remove(page) # Remove the page from its current
position
          stack.append(page) # Push it to the top (most recent)
          results.append('Hit')
      else:
          # Page miss
          misses += 1
          if len(stack) < n:
               stack.append(page) # If there's space in stack
          else:
               stack.pop(0) # Remove the least recently used page
               stack.append(page) # Add the new page
          results.append('Miss')
      # Record the state of the frames
      frame states.append(list(stack))
      # Print the current stack
      print(f"\nCurrent stack: {stack} | Hits: {hits} | Misses:
{misses}")
  return hits, misses, frame_states, results
def main():
  n = int(input("Enter the number of frames (n): "))
  m = int(input("Enter the size of the reference string (m): "))
  reference string = input("Enter the reference string
(space-separated): ").split()
  hits, misses, frame states, results = lru page replacement(n,
reference string)
```

```
print(f"\nTotal Hits: {hits}")
  print(f"Total Misses: {misses}")
  print(f"Hit Ratio: {hits / (hits + misses) if (hits + misses) > 0
else 0:.2f}")
  # Create a DataFrame to visualize the frame states
  df = pd.DataFrame(frame states, columns=[f'Frame {i + 1}' for i
in range(n)])
  df.insert(0, 'Reference String', reference_string)
  df.insert(4, 'Hit/Miss', results) # Insert Hit/Miss column
  print("\nFrame States:")
  print (df)
  print("\n")
  # Plotting the frame states
  fig, ax = plt.subplots(figsize=(10, 3))
  ax.axis('tight')
  ax.axis('off')
  ax.table(cellText=df.values, colLabels=df.columns,
cellLoc='center', loc='center')
  plt.title('Frame States Over Time')
  plt.show()
if <u>__name__</u> == "__main__":
  main()
```

```
Enter the number of frames (n): 3
Enter the size of the reference string (m): 7
Enter the reference string (space-separated): 0 1 2 3 2 1 0
Current stack: ['0'] | Hits: 0 | Misses: 1
Current stack: ['0', '1'] | Hits: 0 | Misses: 2
Current stack: ['0', '1', '2'] | Hits: 0 | Misses: 3
Current stack: ['1', '2', '3'] | Hits: 0 | Misses: 4
Current stack: ['1', '3', '2'] | Hits: 1 | Misses: 4
Current stack: ['3', '2', '1'] | Hits: 2 | Misses: 4
Current stack: ['2', '1', '0'] | Hits: 2 | Misses: 5
Total Hits: 2
Total Misses: 5
Hit Ratio: 0.29
Frame States:
  Reference String Frame 1 Frame 2 Frame 3 Hit/Miss
0
                 0
                         0
                              None
                                       None
                                                Miss
1
                 1
                         0
                                 1
                                       None
                                                Miss
                 2
                         0
                                  1
                                          2
                                                Miss
                 3
                                 2
                                          3
                         1
                                                Miss
4
                 2
                                          2
                         1
                                 3
                                                 Hit
5
                 1
                         3
                                          1
                                  2
                                                 Hit
                 0
                                          0
                                  1
                                                Miss
```

Frame States Over Time

Reference String	Frame 1	Frame 2	Frame 3	Hit/Miss
0	0			Miss
1	0	1		Miss
2	0	1	2	Miss
3	1	2	3	Miss
2	1	3	2	Hit
1	3	2	1	Hit
0	2	1	0	Miss

**Questions:** Solve the same problem manually.

Question:
Implement the Least Recently Used (LRU) page replacement
algorithm for the following parameters:
Number of frames (n): 3
Size of reference string (m): 7
Reference string: '0   2   3   2   1   0'
Solve the problem manually.
Solution:
Step-by-Step Simulation:
Step   Page Requested   Frame State   Hit or Miss   Notes
1 , 0, [0], Miss, Page 0 loaded
2, 1, [0, 1], Miss, Page 1 loaded
3, 2, [0, 1, 2], Miss, Page 2 loaded
4, 3, [1, 2, 3], Miss, Page 0 replaced by 3
5, 2, [1, 2, 3], Hit, Page 2 already in frames
6, 1, [1, 2, 3], Hit, Page 1 already in frames
7, 0, [2, 3, 0], Miss, Page 1 replaced by 0
Summary:
Total Hits: 3
Total Misses: 4
Hit Ratio: 3/7 ≈ 0.43

Outcomes: CO3 – Understand I/O management, memory management and file management

## **Conclusion:** (Conclusion to be based on outcomes achieved)

The Least Recently Used (LRU) page replacement algorithm is essential for effective memory management in operating systems. By retaining the most recently accessed pages, LRU optimizes resource utilization and enhances system performance. Understanding such algorithms is crucial for improving I/O operations and overall system efficiency.

## KJSCE/IT/TY/SEM V/OS/2024-25

Grade: AA / AB / BB / BC / CC / CD / DD

## Signature of faculty in-charge with date

#### **References:**

#### **Books:**

- 1. Applied Operating System Concepts, 1st ed. Silberschatz, Galvin and Gagne, John Wiley Publishers.
- 2. Modern Operating Systems, Tanenbaum, PHI.
- 3. Operating System, 4th Edition, William Stallings, Pearson