

## Lab Assignment 8

**Name:** Manobal Singh Bagady

**SID:** 21104129

**Semester:** 8

**Branch:** Electrical Engineering

**Due Date:** Apr 14, 2025

**Code:**

```
from collections import deque
from prettytable import PrettyTable

class Process:
    def __init__(self, pid, mem_req, page_size):
        self.pid = pid
        self.mem_req = mem_req
        self.page_size = page_size
        self.num_pages = (mem_req + page_size - 1) // page_size
        self.page_table = {i: None for i in range(self.num_pages)}

def simulate_paging(total_mem, frame_size, processes):
    total_frames = total_mem // frame_size
    free_frames = deque(range(total_frames))
    loaded_pages = deque() # FIFO queue of (pid, page)

    for proc in processes:
        print(
            f"\n--- Allocating Process P{proc.pid} ({proc.mem_req}
bytes, {proc.num_pages} pages) ---"
        )
        for page in range(proc.num_pages):
            if free_frames:
                frame = free_frames.popleft()
                proc.page_table[page] = frame
                loaded_pages.append((proc.pid, page))
                print(f"  Page {page} → Frame {frame}")
```

```

        else:
            # FIFO replacement
            old_pid, old_page = loaded_pages.popleft()
            victim = next(p for p in processes if p.pid == old_pid)
            victim_frame = victim.page_table[old_page]
            print(
                f"    [FULL] Evicting P{old_pid}-Pg{old_page} from
Frame {victim_frame}"
            )
            victim.page_table[old_page] = None
            proc.page_table[page] = victim_frame
            loaded_pages.append((proc.pid, page))
            print(f"    Page {page} of P{proc.pid} → Frame
{victim_frame}")

# Build PrettyTable for each process
print("\n=== Final Page Tables ===")
for proc in processes:
    table = PrettyTable()
    table.field_names = ["Page#", "Frame# / Status"]
    for pg, fr in proc.page_table.items():
        status = str(fr) if fr is not None else "Not in Memory"
        table.add_row([pg, status])
    print(f"\nProcess P{proc.pid} Page Table:")
    print(table)

# Build a global frame allocation table
frame_alloc = {f: "Free" for f in range(total_frames)}
for proc in processes:
    for pg, fr in proc.page_table.items():
        if fr is not None:
            frame_alloc[fr] = f"P{proc.pid}-Pg{pg}"

alloc_table = PrettyTable()
alloc_table.field_names = ["Frame#", "Owner"]
for f in range(total_frames):
    alloc_table.add_row([f, frame_alloc[f]])
print("\n=== Frame Allocation ===")
print(alloc_table)

```

```

if __name__ == "__main__":
    total_mem = int(input("Enter total physical memory size (bytes): "))
    frame_size = int(input("Enter frame size (bytes): "))
    n = int(input("Number of processes: "))
    processes = []
    for i in range(1, n + 1):
        mem_req = int(input(f" Process {i} memory requirement (bytes): "))
        processes.append(Process(pid=i, mem_req=mem_req,
page_size=frame_size))
    simulate_paging(total_mem, frame_size, processes)

```

## Output:

```

[legion@LEGION ~]$ python .\Assignment 8.py
Enter total physical memory size (bytes): 400
Enter frame size (bytes): 100
Number of processes: 3
Process 1 memory requirement (bytes): 250
Process 2 memory requirement (bytes): 180
Process 3 memory requirement (bytes): 220

--- Allocating Process P1 (250 bytes, 3 pages) ---
Page 0 → Frame 0
Page 1 → Frame 1
Page 2 → Frame 2

--- Allocating Process P2 (180 bytes, 2 pages) ---
Page 0 → Frame 3
[FULL] Evicting P1-Pg0 from Frame 0
Page 1 of P2 → Frame 0

--- Allocating Process P3 (220 bytes, 3 pages) ---
[FULL] Evicting P1-Pg1 from Frame 1
Page 0 of P3 → Frame 1
[FULL] Evicting P1-Pg2 from Frame 2
Page 1 of P3 → Frame 2
[FULL] Evicting P2-Pg0 from Frame 3
Page 2 of P3 → Frame 3

=== Final Page Tables ===

Process P1 Page Table:
+-----+
| Page# | Frame# / Status |
+-----+
| 0      | Not in Memory   |
| 1      | Not in Memory   |
| 2      | Not in Memory   |
+-----+

```

```

Process P2 Page Table:
+-----+
| Page# | Frame# / Status |
+-----+
| 0      | Not in Memory   |
| 1      | 0                |
+-----+

```

```

Process P3 Page Table:
+-----+
| Page# | Frame# / Status |
+-----+
| 0      | 1                |
| 1      | 2                |
| 2      | 3                |
+-----+

```

```

=== Frame Allocation ===
+-----+
| Frame# | Owner            |
+-----+
| 0      | P2-Pg1           |
| 1      | P3-Pg0           |
| 2      | P3-Pg1           |
| 3      | P3-Pg2           |
+-----+

```