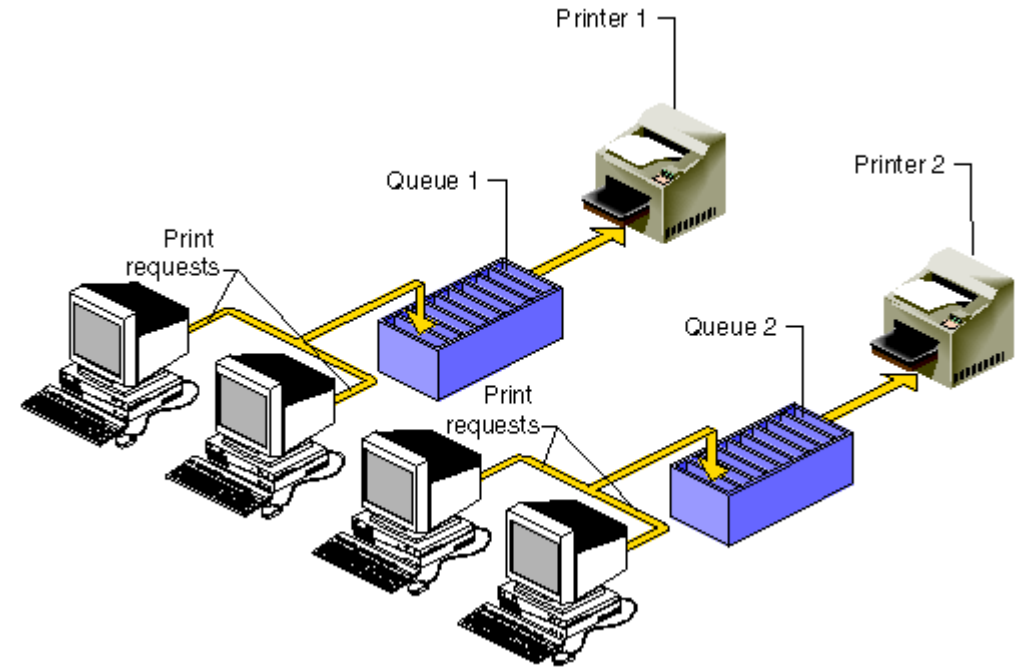


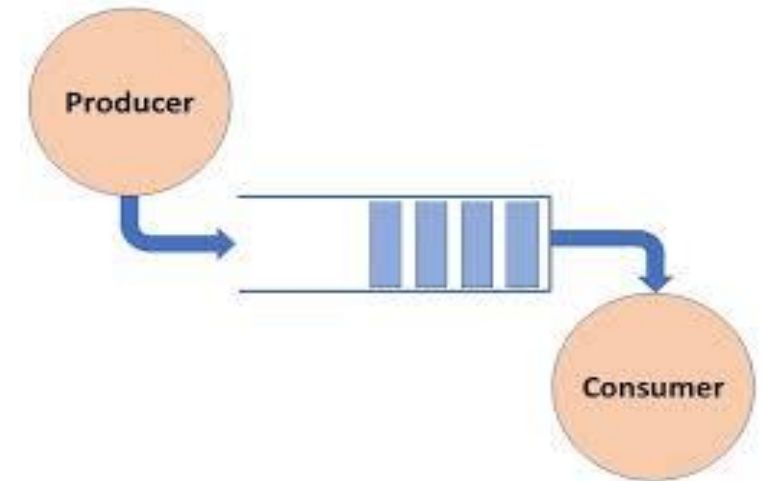
Printer Spooler

The printer spooler, a core OS component, manages print jobs, facilitating efficient printing. It queues tasks, enabling uninterrupted work and network printing. It's essential for synchronization across users and devices, ensuring smooth printing operations. Troubleshooting common issues is crucial. Overall, it enhances printing efficiency, contributing to smoother operations.



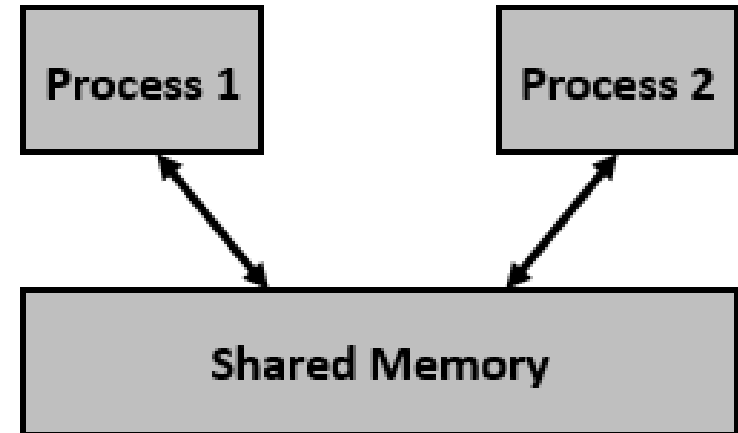
Consumer-Producer Problem

In operating systems, the consumer-producer problem addresses how to coordinate the interaction between entities that generate data (producers) and those that consume it (consumers). This challenge requires ensuring synchronization and preventing issues like data corruption. Solutions involve using synchronization primitives like semaphores and mutexes to control access to shared resources. By managing data flow through strategies like buffer management and signaling mechanisms, operating systems optimize performance and ensure smooth operation.



Memory shared Object

A shared memory object is needed to store the queue buffer and the index of the current entry in this buffer, this is done by sys/mman.h library in c that utilizes shared memory

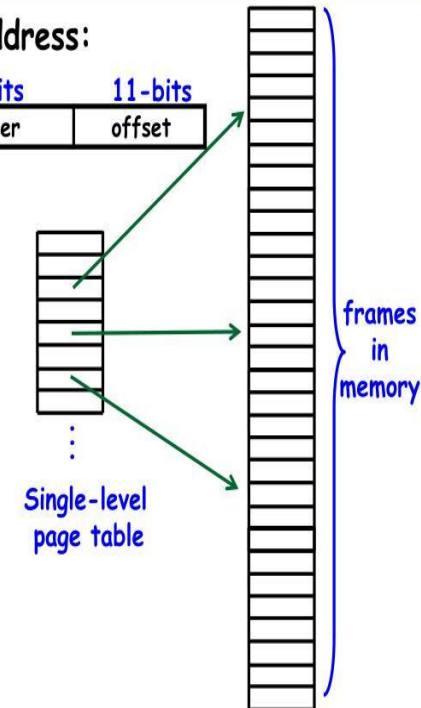


Mapping to Page Table

The mmap function creates a mapping in the process's page table, allowing it to access the shared memory segment at a virtual address, while each process maintains its own independent mapping for efficient communication.

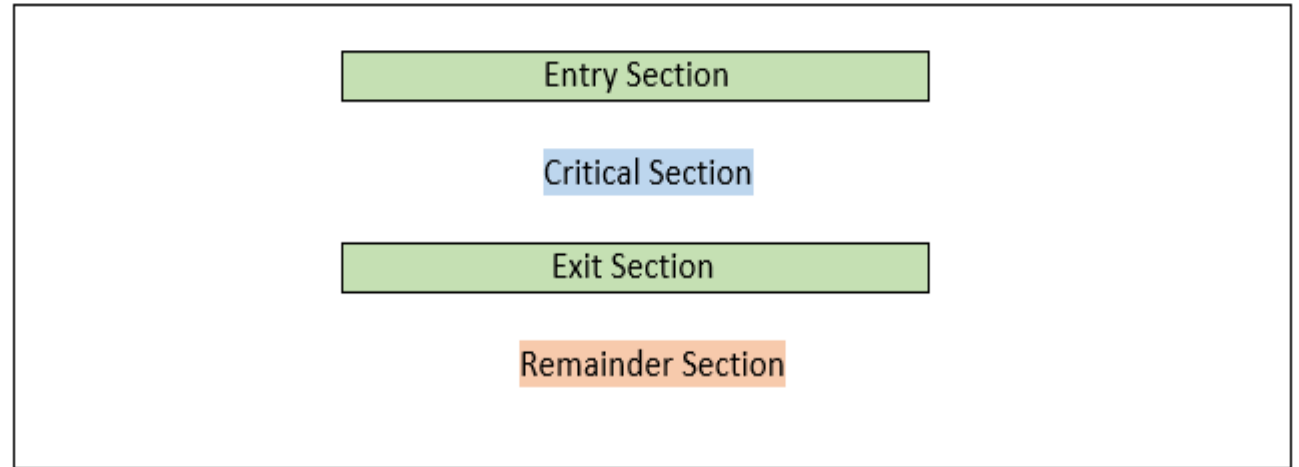
Single-level page tables

A Virtual Address:

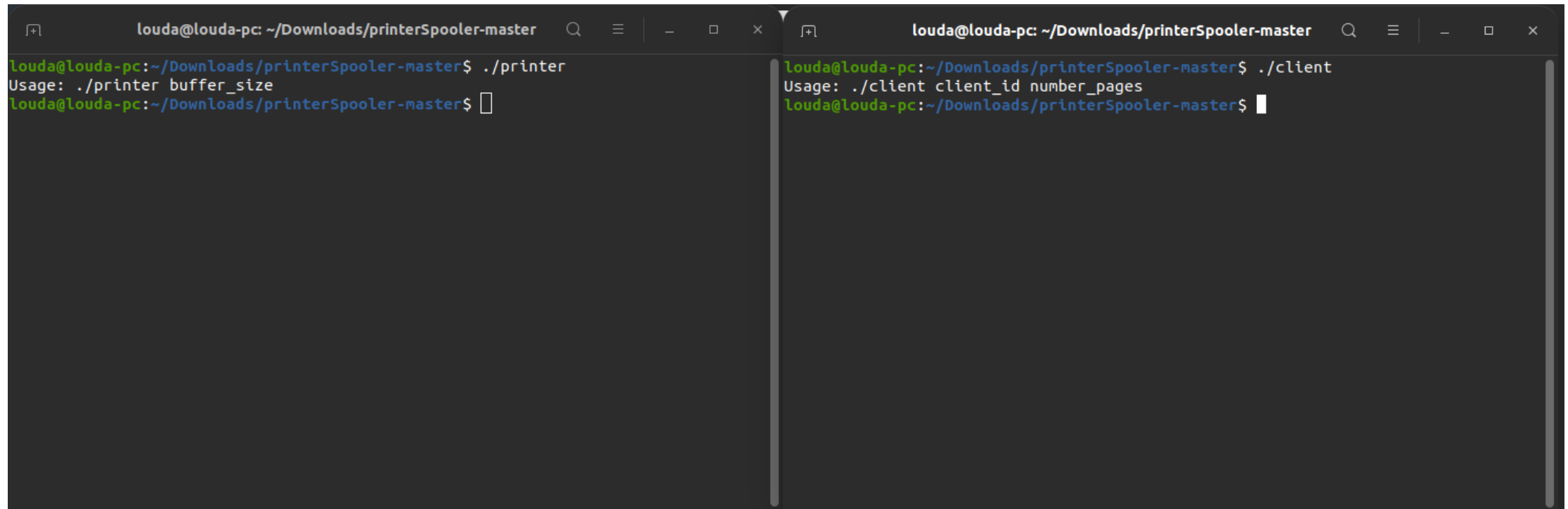


Semaphores

Semaphores are essential for the printer spooler's smooth operation. They act as control gates, similar to traffic lights, ensuring producers only add jobs when the queue has space. Conversely, consumers wait for a semaphore signal indicating a job is available, preventing them from processing an empty queue. This synchronized access by semaphores guarantees data integrity and efficient multi-process interaction within the spooler system.



Program Execution



The image displays two side-by-side terminal windows. Both windows have a title bar that reads 'louda@louda-pc: ~/Downloads/printerSpooler-master'. The left terminal window shows the command './printer' being executed, followed by the usage message 'Usage: ./printer buffer_size'. The right terminal window shows the command './client' being executed, followed by the usage message 'Usage: ./client client_id number_pages'. Both windows have a dark background and light-colored text.

```
louda@louda-pc: ~/Downloads/printerSpooler-master  
louda@louda-pc:~/Downloads/printerSpooler-master$ ./printer  
Usage: ./printer buffer_size  
louda@louda-pc:~/Downloads/printerSpooler-master$  
  
louda@louda-pc: ~/Downloads/printerSpooler-master  
louda@louda-pc:~/Downloads/printerSpooler-master$ ./client  
Usage: ./client client_id number_pages  
louda@louda-pc:~/Downloads/printerSpooler-master$
```

```
lou da@lou da-pc: ~/Downloads/printerSpooler-master  lou da@lou da-pc: ~/Downloads/printerSpooler-master
lou da@lou da-pc:~/Downloads/printerSpooler-master$ ./printer
Usage: ./printer buffer_size
lou da@lou da-pc:~/Downloads/printerSpooler-master$

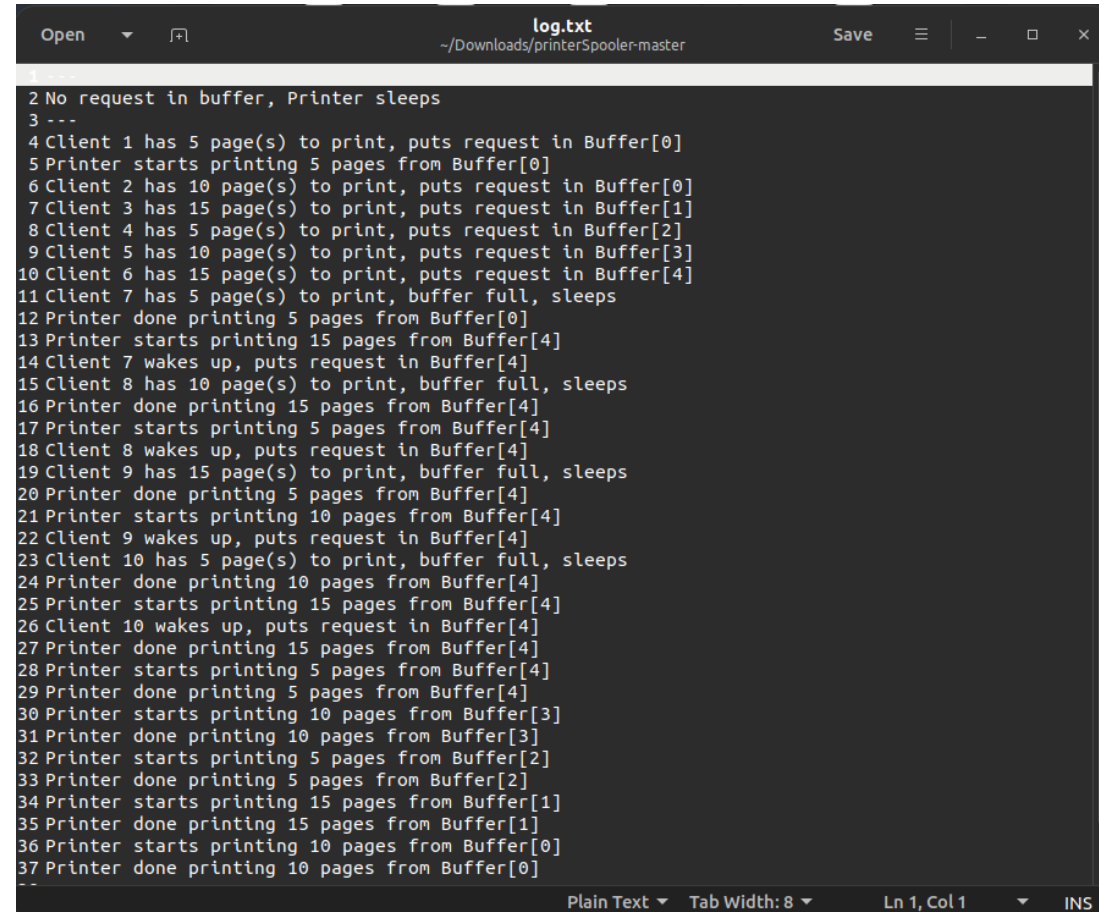
lou da@lou da-pc:~/Downloads/printerSpooler-master$ ./client
lou da@lou da-pc:~/Downloads/printerSpooler-master$ ./client 1 5 && ./client 2 10 &&
lou da@lou da-pc:~/Downloads/printerSpooler-master$ ./client 1 5 && ./client 2 10 &&
./client 3 15 && ./client 4 5 && ./client 5 10 && ./client 6 15 && ./client 7 5 &
& ./client 8 10 && ./client 9 15 && ./client 10 5
shm_open() failed
lou da@lou da-pc:~/Downloads/printerSpooler-master$
```



```
louda@louda-pc: ~/Downloads/printerSpooler-master
Usage: ./printer buffer_size
louda@louda-pc:~/Downloads/printerSpooler-master$ ./printer 5
---
No request in buffer, Printer sleeps
---
Printer starts printing 5 pages from Buffer[0]
Printer done printing 5 pages from Buffer[0]
Printer starts printing 15 pages from Buffer[4]
Printer done printing 15 pages from Buffer[4]
Printer starts printing 5 pages from Buffer[4]
Printer done printing 5 pages from Buffer[4]
Printer starts printing 10 pages from Buffer[4]
Printer done printing 10 pages from Buffer[4]
Printer starts printing 15 pages from Buffer[4]
Printer done printing 15 pages from Buffer[4]
Printer starts printing 5 pages from Buffer[4]
Printer done printing 5 pages from Buffer[4]
Printer starts printing 10 pages from Buffer[3]
Printer done printing 10 pages from Buffer[3]
Printer starts printing 5 pages from Buffer[2]
Printer done printing 5 pages from Buffer[2]
Printer starts printing 15 pages from Buffer[1]
Printer done printing 15 pages from Buffer[1]
Printer starts printing 10 pages from Buffer[0]
Printer done printing 10 pages from Buffer[0]
---
No request in buffer, Printer sleeps
---
```

```
louda@louda-pc:~/Downloads/printerSpooler-master$ ./client
louda@louda-pc:~/Downloads/printerSpooler-master$ ./client 1 5 && ./client 2 10 &&
louda@louda-pc:~/Downloads/printerSpooler-master$ ./client 1 5 && ./client 2 10 &&
./client 3 15 && ./client 4 5 && ./client 5 10 && ./client 6 15 && ./client 7 5 &
& ./client 8 10 && ./client 9 15 && ./client 10 5
shm_open() failed
louda@louda-pc:~/Downloads/printerSpooler-master$ ./client 1 5 && ./client 2 10 &&
./client 3 15 && ./client 4 5 && ./client 5 10 && ./client 6 15 && ./client 7 5 &
& ./client 8 10 && ./client 9 15 && ./client 10 5
Client 1 has 5 page(s) to print, puts request in Buffer[0]
Client 2 has 10 page(s) to print, puts request in Buffer[0]
Client 3 has 15 page(s) to print, puts request in Buffer[1]
Client 4 has 5 page(s) to print, puts request in Buffer[2]
Client 5 has 10 page(s) to print, puts request in Buffer[3]
Client 6 has 15 page(s) to print, puts request in Buffer[4]
Client 7 has 5 page(s) to print, buffer full, sleeps
Client 7 wakes up, puts request in Buffer[4]
Client 8 has 10 page(s) to print, buffer full, sleeps
Client 8 wakes up, puts request in Buffer[4]
Client 9 has 15 page(s) to print, buffer full, sleeps
Client 9 wakes up, puts request in Buffer[4]
Client 10 has 5 page(s) to print, buffer full, sleeps
Client 10 wakes up, puts request in Buffer[4]
louda@louda-pc:~/Downloads/printerSpooler-master$
```

Log File



```
log.txt
~/Downloads/printerSpooler-master

1 ---
2 No request in buffer, Printer sleeps
3 ---
4 Client 1 has 5 page(s) to print, puts request in Buffer[0]
5 Printer starts printing 5 pages from Buffer[0]
6 Client 2 has 10 page(s) to print, puts request in Buffer[0]
7 Client 3 has 15 page(s) to print, puts request in Buffer[1]
8 Client 4 has 5 page(s) to print, puts request in Buffer[2]
9 Client 5 has 10 page(s) to print, puts request in Buffer[3]
10 Client 6 has 15 page(s) to print, puts request in Buffer[4]
11 Client 7 has 5 page(s) to print, buffer full, sleeps
12 Printer done printing 5 pages from Buffer[0]
13 Printer starts printing 15 pages from Buffer[4]
14 Client 7 wakes up, puts request in Buffer[4]
15 Client 8 has 10 page(s) to print, buffer full, sleeps
16 Printer done printing 15 pages from Buffer[4]
17 Printer starts printing 5 pages from Buffer[4]
18 Client 8 wakes up, puts request in Buffer[4]
19 Client 9 has 15 page(s) to print, buffer full, sleeps
20 Printer done printing 5 pages from Buffer[4]
21 Printer starts printing 10 pages from Buffer[4]
22 Client 9 wakes up, puts request in Buffer[4]
23 Client 10 has 5 page(s) to print, buffer full, sleeps
24 Printer done printing 10 pages from Buffer[4]
25 Printer starts printing 15 pages from Buffer[4]
26 Client 10 wakes up, puts request in Buffer[4]
27 Printer done printing 15 pages from Buffer[4]
28 Printer starts printing 5 pages from Buffer[4]
29 Printer done printing 5 pages from Buffer[4]
30 Printer starts printing 10 pages from Buffer[3]
31 Printer done printing 10 pages from Buffer[3]
32 Printer starts printing 5 pages from Buffer[2]
33 Printer done printing 5 pages from Buffer[2]
34 Printer starts printing 15 pages from Buffer[1]
35 Printer done printing 15 pages from Buffer[1]
36 Printer starts printing 10 pages from Buffer[0]
37 Printer done printing 10 pages from Buffer[0]
--
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Team Members

- Ahmed Khaled Ramadan
- Mohamed Sameh Mohamed Mohamed Rezaq
- Omar Ashraf Helmy
- Abdelmoez Ashraf Abdallah