PROBLEM DEFINITION

1. 3 separate processes that talk to each other safely using
2. file operations, shared memory , locks , index based linked list
3. Generator:
    a. Generate and write telemetry events continuously to telemetry.bin (file).
4. Analyzer:
    a. Reads events from telemetry.bin and writes them into shared memory.
    b. Linked list and rolling stats in shared memory.
5. Monitor:
    a. Reads shared memory to compute and print real-time stats.
    b. displays data updated by Analyzer.
6. Shutdown

ALGORITHM –

1 .Generator

continuously generate telemetry events like

event_id ,sensor_id ,timestamp_ns,value ,severity in range 0–5

write and append all events to telemetry.bin
default rate: 100 events per second (configurable)
exit cleanly
on ctrl+C or termination:
stop generating events
close the file

2. Analyzer

initialize shared memory
create shared memory using shm_open.
set the size of shared memory.
map shared memory into process address space.

initialize read write lock and mutex.

initialize event list as empty.

initialize free list with all nodes.

reset all statistics.
read Telemetry File —---->
open telemetry.bin in read mode.
move file pointer to the end.

process events—--->

while shutdown flag is not set:

read one event record from file.

If no new data is available, wait for some time.

take write lock on shared memory.

allocate a free node.
copy event data into the node.
add the node at the end of the list.
increase event count.

If event limit is exceeded >
remove the oldest event from the list.

free that node.
update:

total events count

per sensor count , sum
overall max severity

release write lock.

3. Monitor

Open Shared Memory

Map shared memory into this memory
While shutdown flag is not set
Take Read Lock
Read Total events
Read maximum severity
For each sensor :-
We are printing Average
Release Read lock
Print stats after every 500 milliseconds
Repeat the process continuously

4. Shutdown

On (Ctrl+C):

Set shutdown flag in shared memory (analyzer only).
Generator detects flag and exits loop.
Analyzer detects flag and exits loop.
Monitor detects flag and exits loop.
All processes close files, unmap shared memory, and terminate