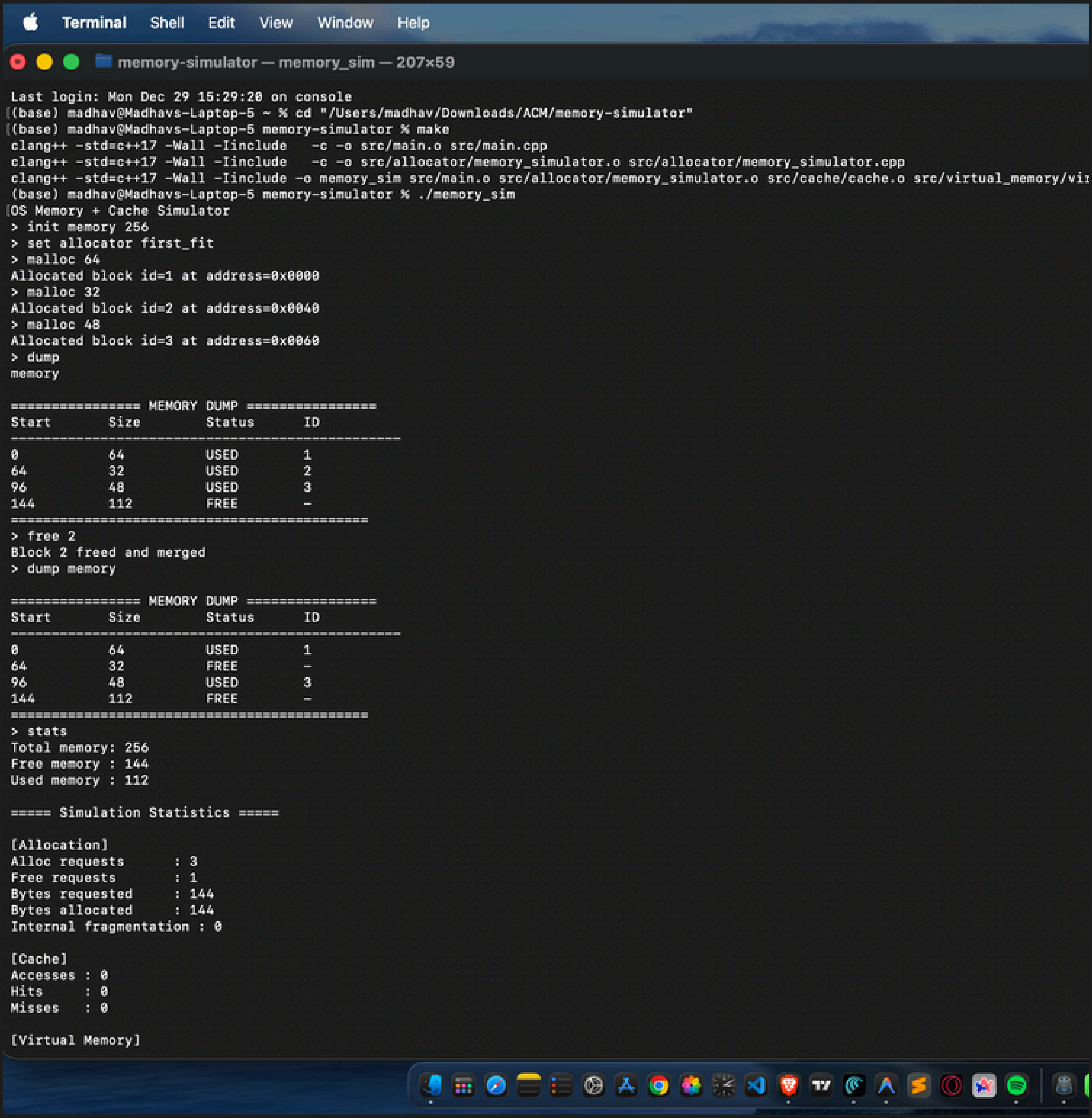
Demonstration document.

Madhav Bansal | 24115094 | EE-2Y

So to demonstrate we have run various test cases on the memory simulator, here are the snapshots demonstrating various test cases.

Test case 1

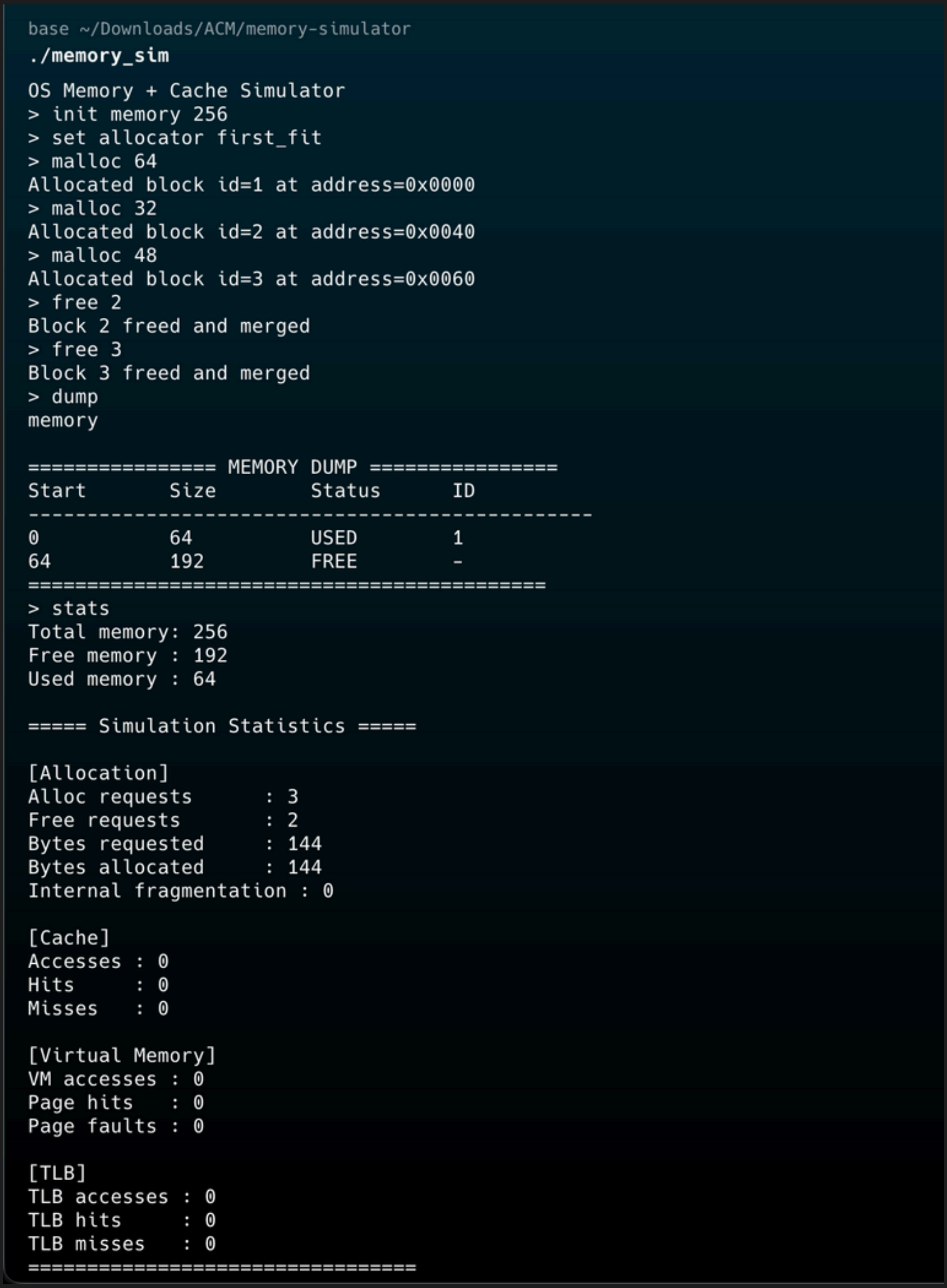
- Demonstrates **First Fit allocation**, where memory blocks are assigned sequentially from the lowest available address and then deallocation of a middle block.
- Memory dump and statistics confirm accurate tracking of used/free memory with zero internal fragmentation.



```
Last login: Mon Dec 29 15:29:20 on console
[(base) madhav@Madhavs-Laptop-5 ~ % cd "/Users/madhav/Downloads/ACM/memory-simulator"
[(base) madhav@Madhavs-Laptop-5 memory-simulator % make
clang++ -std=c++17 -Wall -Iinclude   -c -o src/main.o src/main.cpp
clang++ -std=c++17 -Wall -Iinclude   -c -o src/allocator/memory_simulator.o src/allocator/memory_simulator.cpp
clang++ -std=c++17 -Wall -Iinclude -o memory_sim src/main.o src/allocator/memory_simulator.o src/cache/cache.o src/virtual_memory/vir
(base) madhav@Madhavs-Laptop-5 memory-simulator % ./memory_sim
[OS Memory + Cache Simulator
> init memory 256
> set allocator first_fit
> malloc 64
Allocated block id=1 at address=0x0000
> malloc 32
Allocated block id=2 at address=0x0040
> malloc 48
Allocated block id=3 at address=0x0060
> dump
memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           64          USED        1
64          32          USED        2
96          48          USED        3
144         112         FREE        -
============================================
> free 2
Block 2 freed and merged
> dump memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           64          USED        1
64          32          FREE        -
96          48          USED        3
144         112         FREE        -
============================================
> stats
Total memory: 256
Free memory : 144
Used memory : 112

===== Simulation Statistics =====

[Allocation]
Alloc requests       : 3
Free requests        : 1
Bytes requested      : 144
Bytes allocated      : 144
Internal fragmentation : 0

[Cache]
Accesses : 0
Hits     : 0
Misses   : 0

[Virtual Memory]
```

Test case 2

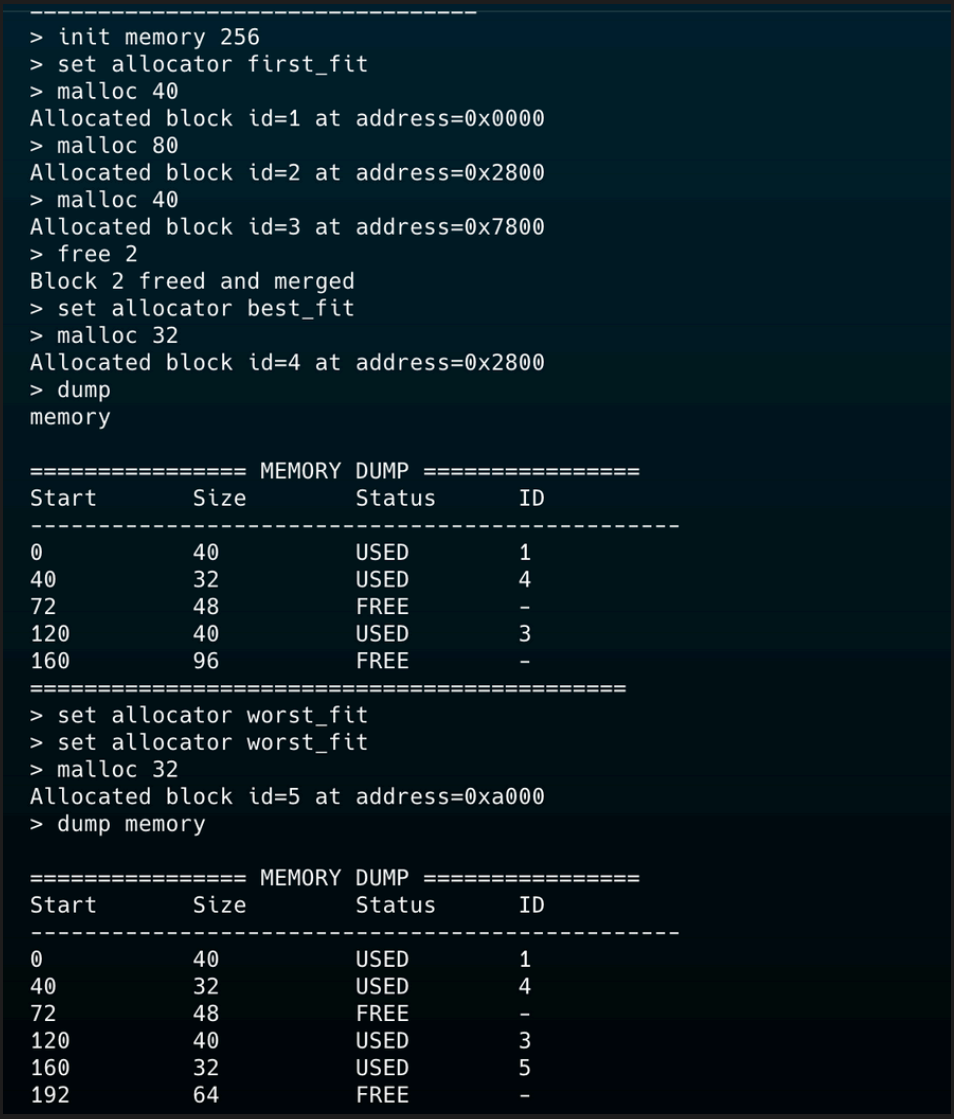
- Demonstrates **coalescing of adjacent free blocks**, where freeing consecutive allocations merges them into a single larger free region which helps in zero external fragmentation.



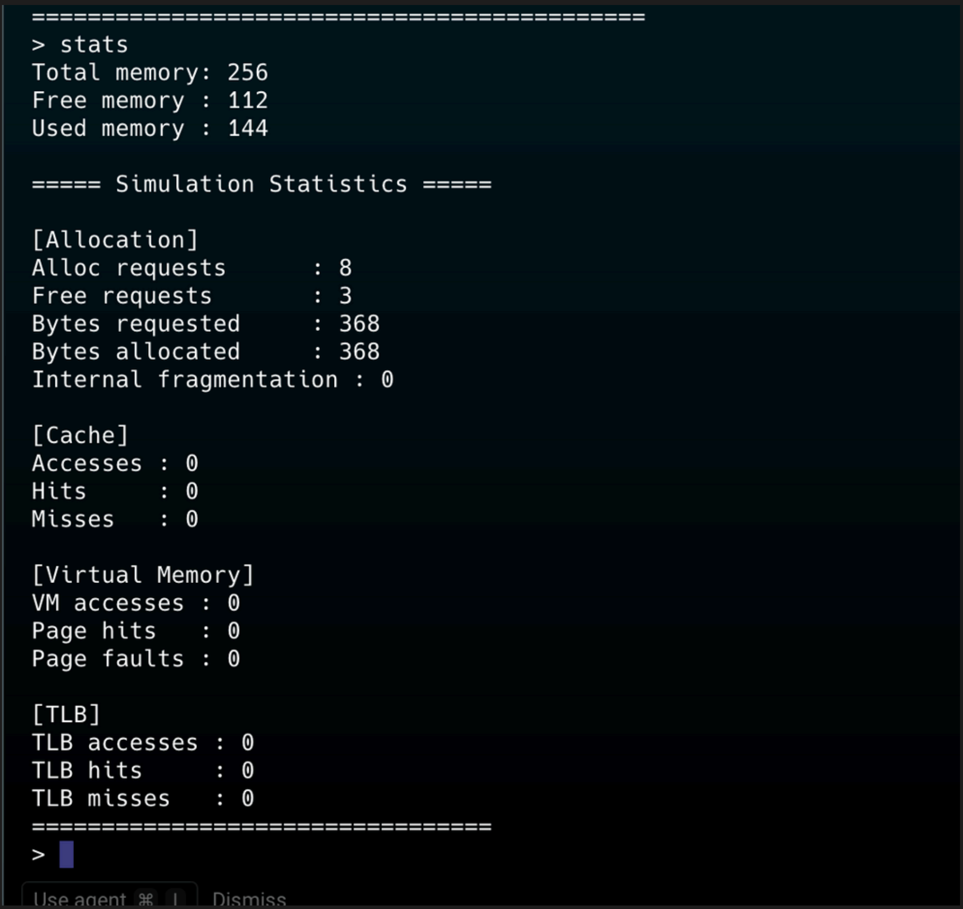
```
base ~/Downloads/ACM/memory-simulator
./memory_sim
OS Memory + Cache Simulator
> init memory 256
> set allocator first_fit
> malloc 64
Allocated block id=1 at address=0x0000
> malloc 32
Allocated block id=2 at address=0x0040
> malloc 48
Allocated block id=3 at address=0x0060
> free 2
Block 2 freed and merged
> free 3
Block 3 freed and merged
> dump
memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           64          USED        1
64          192         FREE        -
============================================
> stats
Total memory: 256
Free memory : 192
Used memory : 64

===== Simulation Statistics =====

[Allocation]
Alloc requests      : 3
Free requests       : 2
Bytes requested     : 144
Bytes allocated     : 144
Internal fragmentation : 0

[Cache]
Accesses : 0
Hits     : 0
Misses   : 0

[Virtual Memory]
VM accesses : 0
Page hits   : 0
Page faults : 0

[TLB]
TLB accesses : 0
TLB hits     : 0
TLB misses   : 0
=================================
```

Test case 3

- Same workload produces different placements under FF, BF and WF such that strategy choice directly impacts free block distribution.



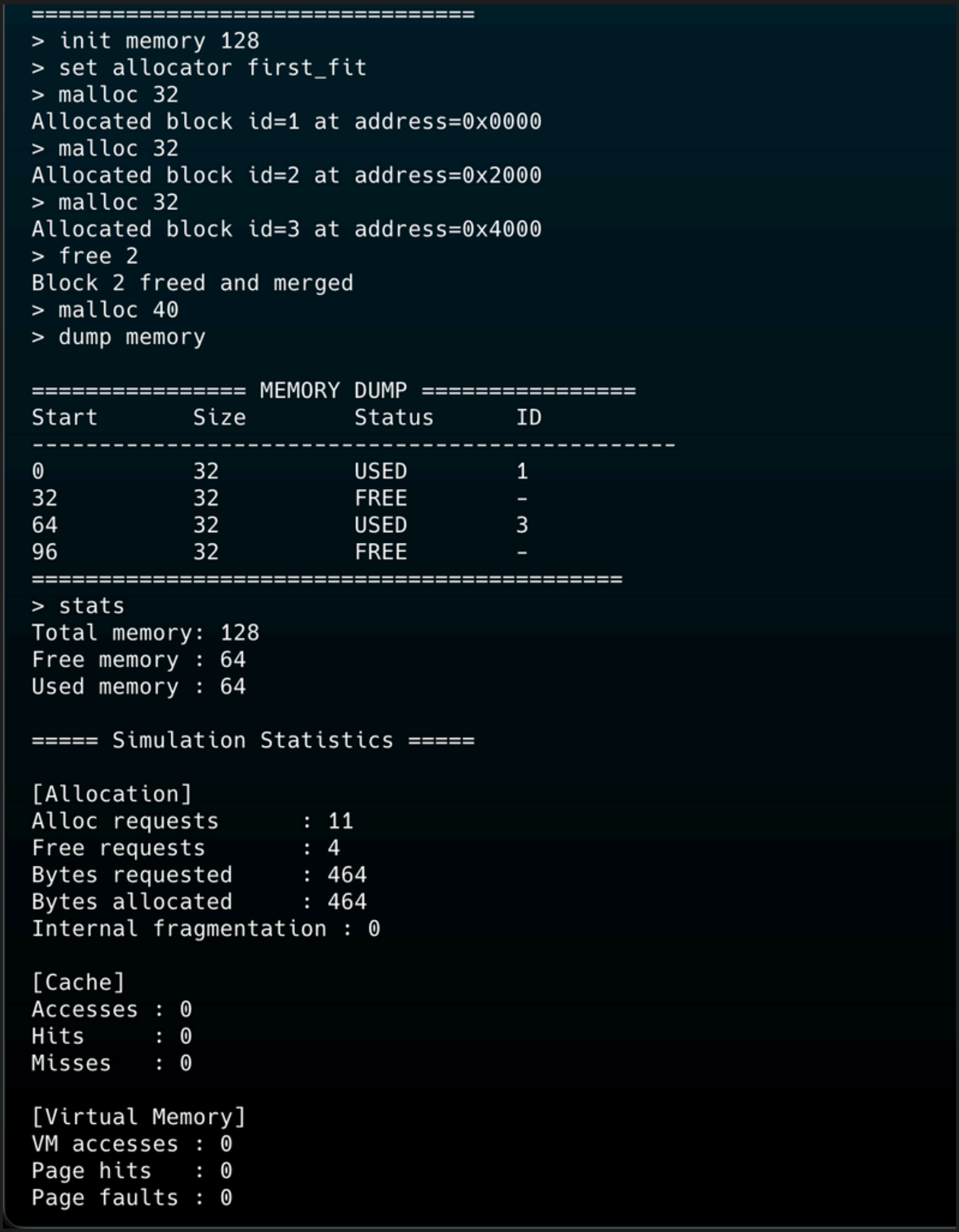
```
---------------------------------
> init memory 256
> set allocator first_fit
> malloc 40
Allocated block id=1 at address=0x0000
> malloc 80
Allocated block id=2 at address=0x2800
> malloc 40
Allocated block id=3 at address=0x7800
> free 2
Block 2 freed and merged
> set allocator best_fit
> malloc 32
Allocated block id=4 at address=0x2800
> dump
memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           40          USED        1
40          32          USED        4
72          48          FREE        -
120         40          USED        3
160         96          FREE        -
============================================
> set allocator worst_fit
> set allocator worst_fit
> malloc 32
Allocated block id=5 at address=0xa000
> dump memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           40          USED        1
40          32          USED        4
72          48          FREE        -
120         40          USED        3
160         32          USED        5
192         64          FREE        -
```



```
=============================================
> stats
Total memory: 256
Free memory : 112
Used memory : 144

===== Simulation Statistics =====

[Allocation]
Alloc requests      : 8
Free requests       : 3
Bytes requested     : 368
Bytes allocated     : 368
Internal fragmentation : 0

[Cache]
Accesses : 0
Hits     : 0
Misses   : 0

[Virtual Memory]
VM accesses : 0
Page hits   : 0
Page faults : 0

[TLB]
TLB accesses : 0
TLB hits     : 0
TLB misses   : 0
=================================
> 
```

Test case 4

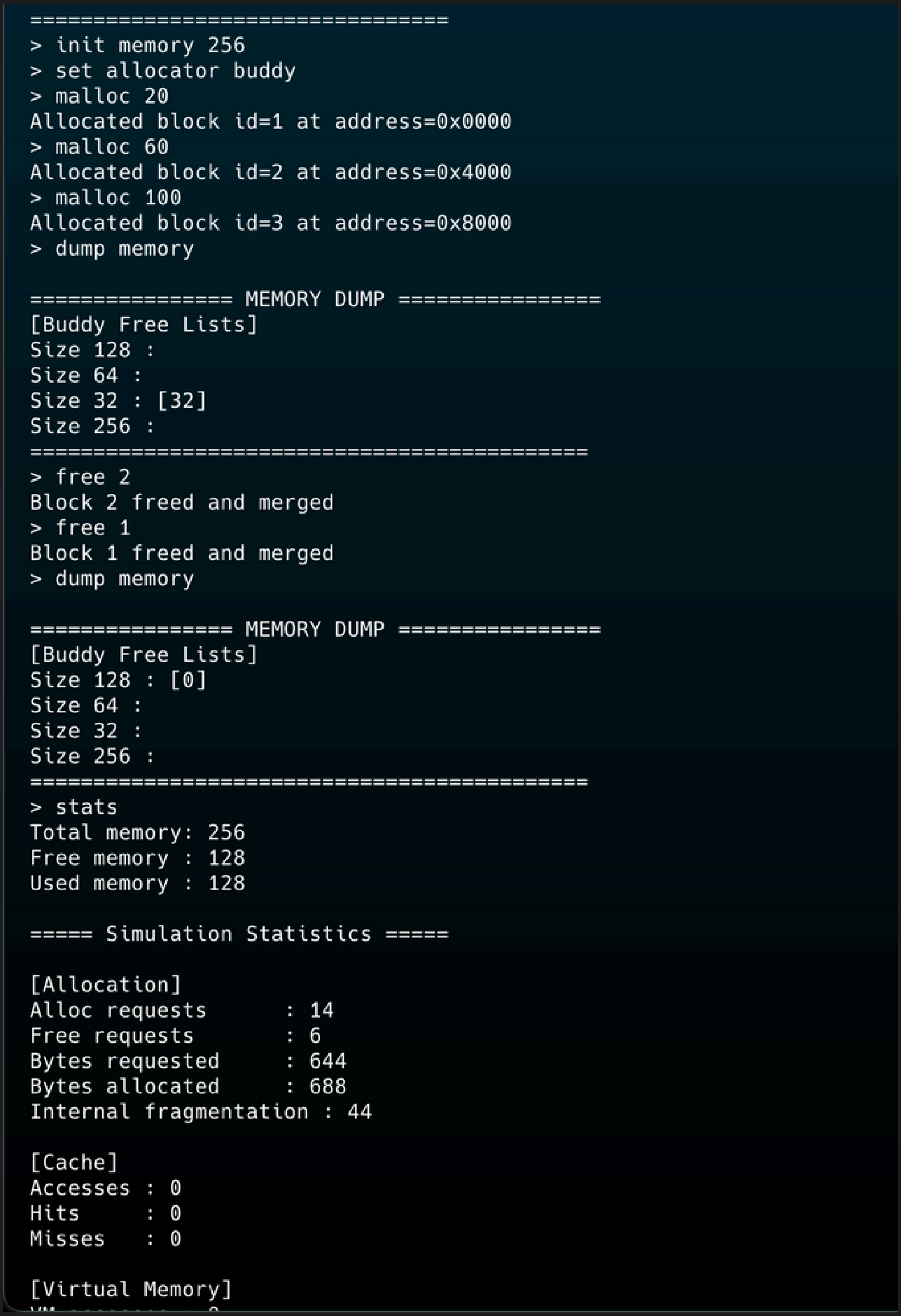
- Demonstrates **external fragmentation**, where sufficient total free memory exists but no single contiguous block can satisfy a larger allocation, dump validates fragmented memory layout.



```
=================================
> init memory 128
> set allocator first_fit
> malloc 32
Allocated block id=1 at address=0x0000
> malloc 32
Allocated block id=2 at address=0x2000
> malloc 32
Allocated block id=3 at address=0x4000
> free 2
Block 2 freed and merged
> malloc 40
> dump memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           32          USED        1
32          32          FREE        -
64          32          USED        3
96          32          FREE        -
============================================
> stats
Total memory: 128
Free memory : 64
Used memory : 64

===== Simulation Statistics =====

[Allocation]
Alloc requests      : 11
Free requests       : 4
Bytes requested     : 464
Bytes allocated     : 464
Internal fragmentation : 0

[Cache]
Accesses : 0
Hits     : 0
Misses   : 0

[Virtual Memory]
VM accesses : 0
Page hits   : 0
Page faults : 0
```

Test case 5

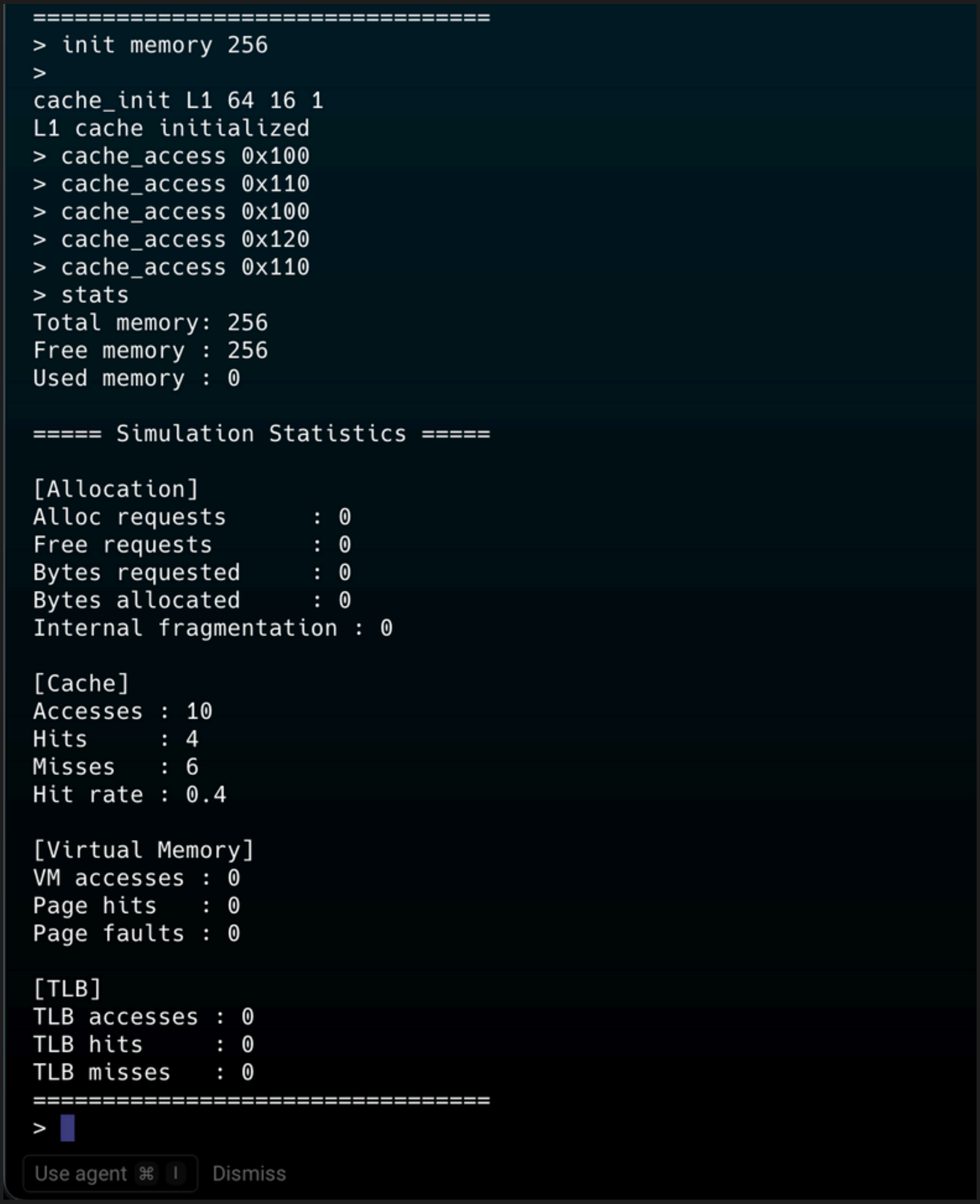
- Demonstrates **buddy allocation**, where requests are rounded up to the nearest power-of-two and blocks are split recursively.
- Shows **buddy coalescing on deallocation**, with freed buddies merging back into larger blocks using address alignment.
- Statistics highlight the trade-off between fast allocation and internal fragmentation inherent to buddy systems.



```
=================================
> init memory 256
> set allocator buddy
> malloc 20
Allocated block id=1 at address=0x0000
> malloc 60
Allocated block id=2 at address=0x4000
> malloc 100
Allocated block id=3 at address=0x8000
> dump memory

================ MEMORY DUMP ================
[Buddy Free Lists]
Size 128 :
Size 64 :
Size 32 : [32]
Size 256 :
============================================
> free 2
Block 2 freed and merged
> free 1
Block 1 freed and merged
> dump memory

================ MEMORY DUMP ================
[Buddy Free Lists]
Size 128 : [0]
Size 64 :
Size 32 :
Size 256 :
============================================
> stats
Total memory: 256
Free memory : 128
Used memory : 128

===== Simulation Statistics =====

[Allocation]
Alloc requests       : 14
Free requests        : 6
Bytes requested      : 644
Bytes allocated      : 688
Internal fragmentation : 44

[Cache]
Accesses : 0
Hits     : 0
Misses   : 0

[Virtual Memory]
VM accesses : 0
```

Test case 6

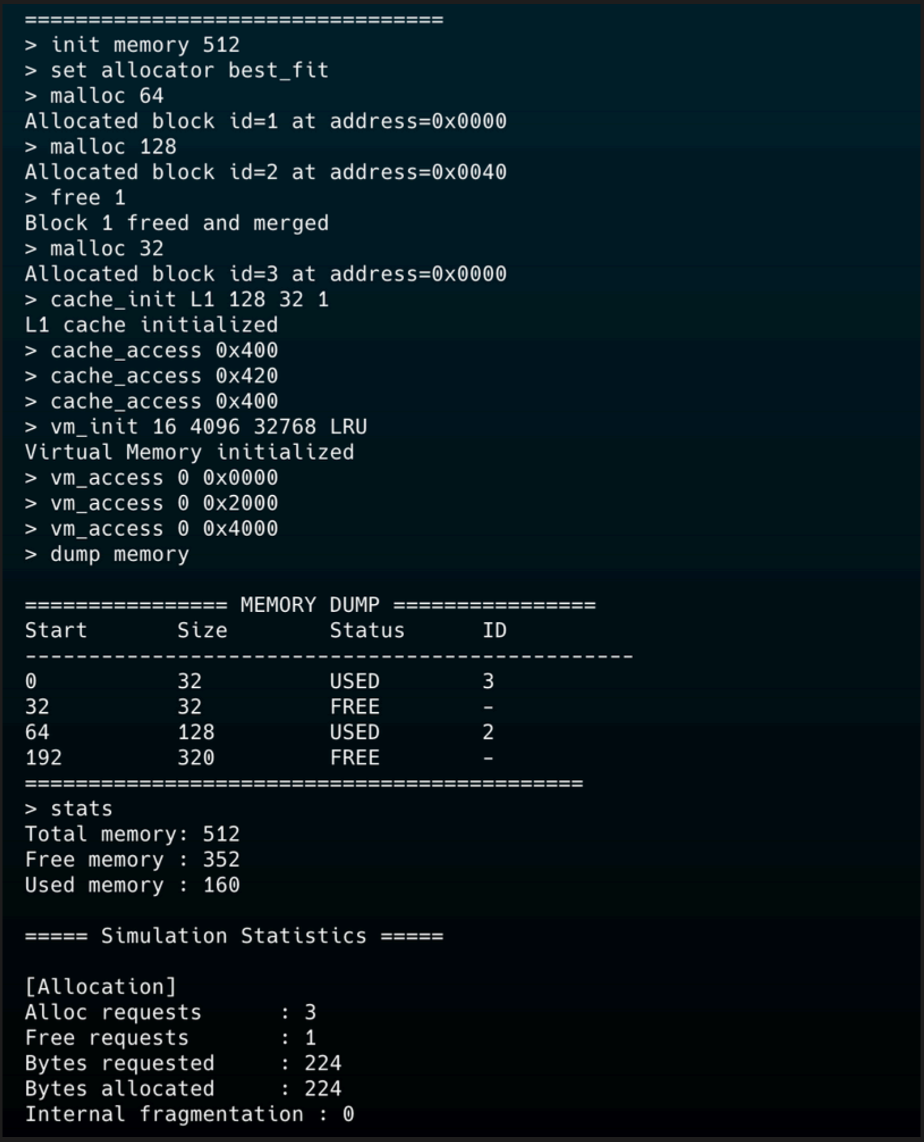
- Demonstrates **cache hit and miss behavior**, including compulsory misses on first access and hits on repeated addresses.
- Confirms correct cache indexing and replacement logic, reflected in consistent hit/miss counts and hit rate.
- Statistics validate **accurate cache performance tracking** independent of memory allocation.



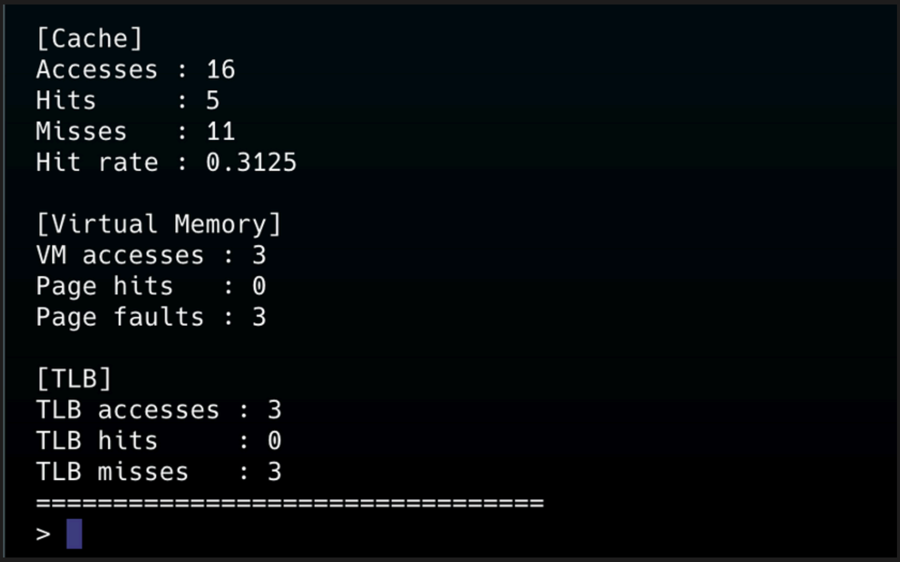
```
=================================
> init memory 256
>
cache_init L1 64 16 1
L1 cache initialized
> cache_access 0x100
> cache_access 0x110
> cache_access 0x100
> cache_access 0x120
> cache_access 0x110
> stats
Total memory: 256
Free memory : 256
Used memory : 0

===== Simulation Statistics =====

[Allocation]
Alloc requests      : 0
Free requests       : 0
Bytes requested     : 0
Bytes allocated     : 0
Internal fragmentation : 0

[Cache]
Accesses : 10
Hits     : 4
Misses   : 6
Hit rate : 0.4

[Virtual Memory]
VM accesses : 0
Page hits   : 0
Page faults : 0

[TLB]
TLB accesses : 0
TLB hits     : 0
TLB misses   : 0
=================================
>
```

Test case 7

- Demonstrates integrated operation of memory allocation, cache access and **virtual memory translation** within a single execution flow.
- Shows cache behavior under real address streams and **page faults on first-time** virtual page accesses, validating correct VM–cache interaction



```
=================================
> init memory 512
> set allocator best_fit
> malloc 64
Allocated block id=1 at address=0x0000
> malloc 128
Allocated block id=2 at address=0x0040
> free 1
Block 1 freed and merged
> malloc 32
Allocated block id=3 at address=0x0000
> cache_init L1 128 32 1
L1 cache initialized
> cache_access 0x400
> cache_access 0x420
> cache_access 0x400
> vm_init 16 4096 32768 LRU
Virtual Memory initialized
> vm_access 0 0x0000
> vm_access 0 0x2000
> vm_access 0 0x4000
> dump memory

================ MEMORY DUMP ================
Start       Size        Status      ID
------------------------------------------------
0           32          USED        3
32          32          FREE        -
64          128         USED        2
192         320         FREE        -
============================================
> stats
Total memory: 512
Free memory : 352
Used memory : 160

===== Simulation Statistics =====

[Allocation]
Alloc requests      : 3
Free requests       : 1
Bytes requested     : 224
Bytes allocated     : 224
Internal fragmentation : 0
```



```
[Cache]
Accesses : 16
Hits     : 5
Misses   : 11
Hit rate : 0.3125

[Virtual Memory]
VM accesses : 3
Page hits   : 0
Page faults : 3

[TLB]
TLB accesses : 3
TLB hits     : 0
TLB misses   : 3
=================================
>
```