OS’23 Project

TESTING WHOLE PROJECT SHORT NOTES

Agenda

- Round Robin (RR)
- BSD Scheduler

Round Robin (RR)

FIFO STRATEGY

Testing Whole Project – Scenario 1

Running single program **NO** PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (malloc & free)
3. Page Fault Handler (placement [**only**])

Scenario Sequence (DO the following to TEST this scenario):

FOS> run qs 3000

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **5,000**
  Initialization method : **Descending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **300,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 2

Running single program with PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap

2. USER Heap (malloc & free)

3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

FOS> run qs 7

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **5,000**
  Initialization method : **Descending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **300,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 3
Running multiple programs with **NO** PAGES suffocation (1)

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (malloc ONLY)
3. Page Fault Handler (placement [**Only**])

Scenario Sequence (DO the following to TEST this scenario):

1. FOS> load fib 300 //load Fibonacci program
2. FOS> load qs 3000 //load Quick sort program [with leakage]
3. FOS> load ms2 1500 //load Merge sort program [with leakage]
4. FOS> runall //run all of them together

Test them according to the following steps:

[Fibonacci]

Fibonacci index = **30 "Result should = 1346269"**

Testing Whole Project – Scenario 3
Running multiple programs with **NO** PAGES suffocation (2)

[QuickSort]

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **1,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

[MergeSort]

- Number of Elements = **32**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **32**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 4

Running multiple programs with PAGES suffocation (1)

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (malloc ONLY)
3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

1. FOS> load fib 7 //load Fibonacci program
2. FOS> load qs 7 //load Quick sort program [with leakage]
3. FOS> load ms2 7 //load Merge sort program [with leakage]
4. FOS> runall //run all of them together

Test them according to the following steps:

[Fibonacci]

Fibonacci index = **30 "Result should = 1346269"**

Testing Whole Project – Scenario 4

Running multiple programs with PAGES suffocation (2)

[QuickSort]

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **1,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

[MergeSort]

- Number of Elements = **32**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **32**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 5
Effect of Memory Leakage!! with **NO** PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (Efficiency of First Fit)
3. Page Fault Handler (placement [**Only**])

Scenario Sequence (DO the following to TEST this scenario):

FOS> run ms1 1500 //run Merge sort with NO memory leakage

Number of Elements = **5,000**
Initialization method : **Ascending**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

FOS> run ms2 1500 //run Merge sort that CAUSE memory leakage

Number of Elements = **5,000**
Initialization method : **Semi random**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

WHAT Happens?! WHY?

Testing Whole Project – Scenario 6

Effect of Memory Leakage!! with PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (Efficiency of First Fit)
3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

FOS> run ms1 7 //run Merge sort with NO memory leakage

Number of Elements = **5,000**
Initialization method : **Ascending**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

FOS> run ms2 7 //run Merge sort that CAUSE memory leakage

Number of Elements = **5,000**
Initialization method : **Semi random**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

WHAT Happens?! WHY?

LRU LISTS STRATEGY

Testing Whole Project – Scenario 1

Running single program **NO** PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap

2. USER Heap (malloc & free)

3. Page Fault Handler (placement [**only**])

Scenario Sequence (DO the following to TEST this scenario):

FOS> lru 2

FOS> run qs 3000 500

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **5,000**
  Initialization method : **Descending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **300,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 2

Running single program with PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap

2. USER Heap (malloc & free)

3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

FOS> lru 2

FOS> run qs 7 3

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **5,000**
  Initialization method : **Descending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **300,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 3
Running multiple programs with **NO** PAGES suffocation (1)

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (malloc ONLY)
3. Page Fault Handler (placement [**Only**])

Scenario Sequence (DO the following to TEST this scenario):

1. FOS> lru 2
2. FOS> load fib 300 50 //load Fibonacci program
3. FOS> load qs 3000 500 //load Quick sort program [with leakage]
4. FOS> load ms2 1500 500 //load Merge sort program [with leakage]
5. FOS> runall //run all of them together

Test them according to the following steps:

[Fibonacci]

Fibonacci index = **30 "Result should = 1346269"**

Testing Whole Project – Scenario 3
Running multiple programs with **NO** PAGES suffocation (2)

[QuickSort]

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **1,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

[MergeSort]

- Number of Elements = **32**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **32**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 4

Running multiple programs with PAGES suffocation (1)

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (malloc ONLY)
3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

1. FOS> lru 2
2. FOS> load fib 7 2 //load Fibonacci program
3. FOS> load qs 7 2 //load Quick sort program [with leakage]
4. FOS> load ms2 7 2 //load Merge sort program [with leakage]
5. FOS> runall //run all of them together

Test them according to the following steps:

[Fibonacci]

Fibonacci index = **30 "Result should = 1346269"**

Testing Whole Project – Scenario 4

Running multiple programs with PAGES suffocation (2)

[QuickSort]

- Number of Elements = **1,000**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **1,000**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

[MergeSort]

- Number of Elements = **32**
  Initialization method : **Ascending**
  Do you want to repeat (y/n): **y**

- Number of Elements = **32**
  Initialization method : **Semi random**
  Do you want to repeat (y/n): **n**
  "At each step, the program should sort the array successfully"

Testing Whole Project – Scenario 5
Effect of Memory Leakage!! with **NO** PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (Efficiency of First Fit)
3. Page Fault Handler (placement [**Only**])

Scenario Sequence (DO the following to TEST this scenario):

FOS> lru 2
FOS> run ms1 1500 500 //run Merge sort with NO memory leakage

Number of Elements = **5,000**
Initialization method : **Ascending**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

FOS> lru 2

FOS> run ms2 1500 500 //run Merge sort that CAUSE memory leakage

Number of Elements = **5,000**
Initialization method : **Semi random**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

WHAT Happens?! WHY?

Testing Whole Project – Scenario 6

Effect of Memory Leakage!! with PAGES suffocation

REQUIRED MODULES:

1. KERNEL Heap
2. USER Heap (Efficiency of First Fit)
3. Page Fault Handler (placement + replacement)

Scenario Sequence (DO the following to TEST this scenario):

FOS> lru 2
FOS> run ms1 7 2 //run Merge sort with NO memory leakage

Number of Elements = **5,000**
Initialization method : **Ascending**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

FOS> run ms2 7 2 //run Merge sort that CAUSE memory leakage

Number of Elements = **5,000**
Initialization method : **Semi random**
Do you want to repeat (y/n): **n**
"the program should sort the array successfully"

WHAT Happens?! WHY?

Thank you for your care…

Enjoy **making** your **own FOS** ☺️