Safe GC

Manavjeet Singh

2018295

How did you find the object header corresponding to a heap address (including big allocations)?

- First check if the address lies in the heap segment
- Using segment, check if BigAlloc is 1 or 0, for big allocations and small allocations respectively.
- For Small Allocations:
  - Calculate starting of the page in which the heap address lies
  - Traverse all the objects present on the page and until an object is found whose base_address+OBJ_HEADER_SIZE is less or equal to the Heap_address and base_address+size_of_object> Heap_address
  - If the above condition is true then base_addr is the required Object header.
- For Big Allocations
  - Calculate current page using the given heap address
  - While free space in current_page (check from metadata) is not equal to 1, current_page-=PAGE_SIZE
  - When such page is found, return the address.

Discuss your implementation of sweep.

While iterating on heap at an offset of 4-bytes

- while in the list of segments:
  - Check if BigAlloc is 1 or 0 for the present segment
  - For Big Allocations
    - if free space on the page==PAGE_SIZE, do nothing
    - if segment→Status!=FREE or segment→Status !=MARK
      - free the segment using myfree
  - For Small Allocations

- let temp=(object header *)data_pointer of the current segment
- while temp<alloc_pointer of that segment
  - if free space on the current page==PAGE_SIZE, increase temp to next page
  - if temp→Status!=FREE or temp→Status !=MARK
    - free temp using myfree
  - increase temp by temp→Size, but if temp goes to next page, then temp=starting of that page.

Did you add any new “struct”

Added new linked list structure for scanner list.

```
typedef struct Scanner
{
     ObjHeader* addr;
     struct Scanner *next;
} ScannerList;

ScannerList *scannerlist_start;
ScannerList *scannerlist_end;
```

The output of “make run”

```
/usr/bin/time -v ./random
total edges:4222800
Num Bytes Allocated: 476000016
Num Bytes Freed: 27652752
Num GC Triggered: 14
printing stats after final GC
Num Bytes Allocated: 476000016
Num Bytes Freed: 28345248
Num GC Triggered: 15
     Command being timed: "./random"
```

```
User time (seconds): 9.51
System time (seconds): 0.29
Percent of CPU this job got: 99%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:09.81
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 723992
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 180753
Voluntary context switches: 1
Involuntary context switches: 51
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```