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.

- Traversing in segment list, and for each segment:
  1. get data pointers and alloc pointers of the segment
  2. create a pointer named temp of type ObjHeader, pointing to the data pointer
  3. check from metadata of the current page has free space less than PAGE_SIZE, if not, increment temp to next page and repeat this step.
  4. free the temp object, if Status is not free and not marked.
  5. Increment temp pointer to point memory location temp->Size ahead.
  6. if the temp is less than alloc pointer then go to step3
- Unmark all the objects of the Scanner list.

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;
```

- I also added a constant INLIST to mark if a Object is added to scanner list in object header itself.

The output of “make run”

```
/usr/bin/time -v ./random
total edges:4222800
Num Bytes Allocated: 476000016
Num Bytes Freed: 425764944
Num GC Triggered: 14
printing stats after final GC
Num Bytes Allocated: 476000016
Num Bytes Freed: 475200000
Num GC Triggered: 15
     Command being timed: "./random"
     User time (seconds): 12.92
     System time (seconds): 0.49
     Percent of CPU this job got: 100%
     Elapsed (wall clock) time (h:mm:ss or m:ss): 0:13.42
     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): 187220

Average resident set size (kbytes): 0

Major (requiring I/O) page faults: 0

Minor (reclaiming a frame) page faults: 128764

Voluntary context switches: 1

Involuntary context switches: 68

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