15-213: Introduction to Computer Systems

Written Assignment #7

This written homework covers Dynamic Memory Allocation.

Directions

Complete the question(s) on the following pages with single paragraph answers. These questions are not meant to be particularly long! Once you are done, submit this assignment on Canvas. Below is an example question and answer.

Q: In a few sentences, give two practical reasons why two's complement signed integers are more convenient to work with than one's complement or sign-and-magnitude.

A: Other representations of signed integers (ones-complement and sign-and-magnitude) have two representations of zero (+0 and −0), which makes testing for a zero result more difficult. Also, addition and subtraction of two's complement signed numbers are done exactly the same as addition and subtraction of unsigned numbers (with wraparound on overflow), which means a CPU can use the same hardware and machine instructions for both.

Grading

Each assignment will be graded in two parts:

1. Does this work indicate any effort? (e.g. it's not copied from a homework for another class or from the book)
2. Three peers will provide short, constructive feedback.

This assignment is due on _____,_____, 2022 by 11:59pm Pittsburgh time (currently UTC−4). Remember to convert this time to the timezone you currently reside in.

Question #1

A. Identify the kind of fragmentation in the below scenarios

Scenario 1: Requested Payload is 8 bytes but the block size is 32 bytes
Scenario 2: Requested Payload is 64 bytes Total Free memory available is 64 bytes but it is not available in one contiguous chunk.

Scenario 1: Internal fragmentation
Scenario 2: External fragmentation

B. What is the difference between internal fragmentation and external fragmentation? Is there a way to decrease the amount of internal fragmentation? What about external fragmentation? Answer why or why not for each.

Internal fragmentation is when there is unused memory inside of a block of malloced space.
External fragmentation is when there is unused memory between blocks of malloced space.
Both are unused spaces in memory, though they represent different spaces (inside of a currently malloced, or currently free).

We can reduce internal fragmentation by changing our representations of the free list, either through encoding information in unused bits or reducing the size of our free list nodes.

full points should be awarded for explaining that external fragmentation can be decreased by coalescing or using a best-fit algorithm.

Question #2

A memory allocator (malloc) has the following properties.

- At the start, there is a prologue and an epilogue (each 8 bytes).
- Each block has 8 byte headers and 8 byte footers.
- The blocks of memory are 16-byte aligned.
- Blocks are coalesced if possible

A. Some statistics for a set of trace files indicate that the best balance between throughput and utilization is achieved when the chunk size is 2048 bytes. How much should the heap expand by when there is no space available for a new request? Try to think about different cases!

Answer:
If the requested payload size is lesser than chunk size,
    expand by chunk size
If the requested payload size is greater than the chunk size,

it would be smarter to expand by the payload size (rounded up to maintain alignment)

B. The below code is executed, draw out how the heap looks after executing every free request.

```
void *block1 = malloc(43);
void *block2 = malloc(5);
free(block1);
void *block3 = malloc(1198);
void *block4 = malloc(515);
void *block5 = malloc(7010);
free(block4);
```

Answer:

It is mentioned that we only have the prologue and epilogue with each being 8 bytes. Before we get to the malloc part, the mm_init function gets called and this calls the extend heap function with the chunksize(2048) and this is where our free pointer points to initially.

Prologue (8 bytes)	Free Block (2048 bytes)	Epilogue (8 bytes)

a. block1 -> malloc(43)
Since 43 bytes are requested, total would be 43+8(header)+8(footer) = 59 is needed. So round off to 64 bytes.

Prologue (8 bytes)	Block 1 (64 bytes)	Free Block (1984 bytes)	Epilogue (8 bytes)

b. block2 -> malloc(5)
Total needed with headers = 5+16=21 rounded off to 32 bytes.

P (8)	B1 (64)	B2 (32)	Free (1952)	E (8 bytes)

c. free(block1);
This requires us to free the allocated block 1(B1).

P (8)	Free (64)	B2 (32)	Free (1952)	E (8 bytes)

d. block3-> malloc(1198);
Total needed=1198+16(Header, Footer)=1214. So round off to 1216.

P (8)	Free (64)	B2 (32)	B3 (1216)	Free (736)	E (8 bytes)

d. block4-> malloc(515);
Total needed=515+16=531 rounded off to 544 bytes.

P (8)	Free (64)	B2 (32)	B3 (1216)	B4 (544)	Free (192)	E (8 bytes)

d. block5-> malloc(7010);
Here the requested amount is greater than the chunksize. So the closest 16 aligned value to 7010+16=7026 which is 7040 bytes.

P (8)	Free (64)	B2 (32)	B3 (1216)	B4 (544)	B5 (7040)	Free (192)	E (8 bytes)

e. free(block4);
This requires us to free the allocated block 4(B4).

P (8)	Free (64)	B2 (32)	B3 (1216)	Free (544)	B5 (7040)	Free (192)	E (8 bytes)

Question #3

Given the following code snippet:

```
1      int main()
2      {
3         if (fork())
4         {
5            printf("ka");
6            waitpid(-1, NULL, 0);
7         }
8         else
9         {
10           printf("pi");
11           exit(0);
12        }
13        printf("chu");
14        exit(0);
15     }
```

Assume that all system calls succeed and error checking has been omitted.

1. What are all the possible output sequences?
2. If we remove exit(0); in line 11, now what are all the possible output sequences?

Answer
a.
pikachu,
kapichu

b.
pichukachu,
kapichuchu,
pikachuchu

Summer 2022 Written Assignment # Rubric

Criteria	Ratings	Pts

	Question #1 Correctly identifies the difference between internal fragmentation and external fragmentation	1 for Full Pts	0 pts No Marks	
	Question #1 Explains whether it is possible decrease the amount of internal fragmentation	1 for Full Pts	0 pts No Marks	
	Question #1 Explains whether it is possible decrease the amount of external fragmentation	1 for Full Pts	0 pts No Marks	
	Question #2 Correctly States that if the request payload size is less than the chunksize, we can extend the chunksize, if the payload size is larger than the chunksize, we should expand the payload size(with alignment)	1 for Full Pts	0 pts No Marks	
	Question #2 Correctly identifies the heap structure after the second free.	1 for Full Pts	0 pts No Marks	
	Question #3 1point for a (0.5 for answer and 0.5 for process graph), 1 point for b(0.5 for answer and 0.5 for process graph). Partial points for partially correct answers.	2 for Full Pts	0 pts No Marks	