Assignment 3
Malloclab

Computer Security Lab

Name: Suhwan Song

Email: sshkeb96@snu.ac.kr

Malloclab Goal

- Implement an allocator that is correct, efficient and fast.

Malloclab Overview

- You should implement one of four allocators:
  - Implicit list
  - Explicit list
  - Segregated free list
  - Blocks sorted by size

- For evaluation:
  - 11 trace files are provided.
  - You will get perfect score when implementing the correct, fast, and memory-efficient allocator.

Downloading Your Malloclab

- You can download malloclab-handout.tar at **eTL**.

How to copy malloclab-handout.tar into your container

- copy host file to the container

```
> docker cp malloclab-handout.tar <CONTAINER_NAME>:<DESTINATION_PATH>
```

- In docker container, extract the bufbomb

```
(in docker) > tar xvf malloclab-handout.tar
```

How to Work on the Lab

- malloclab-handout has 2 main files
    - **mm.c**: Your solution malloc file that you should modify.
    - **mdriver.c**: The malloc driver that tests your mm.c files.

- Your dynamic storage allocator will consist of the following four functions, which are defined in **mm.c**.
    - int mm_init(void);
    - void *mm_malloc(size_t size);
    - void mm_free(void *ptr);
    - void *mm_realloc(void *ptr, size_t size); // please refer to malloclab-readme.pdf

How to Work on the Lab

- int mm_init(void);
  - Performs any necessary initializations, such as allocating the initial heap area.
  - Returns **-1** if there was a problem in performing the initialization, **0** otherwise.

- void *mm_malloc(size_t size);
  - Returns a pointer (**8-byte aligned pointers)** to an allocated block payload of at least *size* bytes.
  - Returns **NULL** if there was a problem in performing the allocation.
  - The entire allocated block should lie within the heap region and should not overlap with others.

- void mm_free(void *ptr);
  - Frees the block pointed to by ptr.
  - Returns nothing.

How to Test Your Implementation

- To build mdriver with your source code file mm.c.

```
> make
```

- To test your code with a rep file.

```
> ./mdriver -V -f {rep file}
```

- For evaluation,

```
> ./mdriver -V -t ./traces
```

How to Submit

- Prepare for submit. This will generate **assign3.tar.gz** file.

```
> ./prepare_for_submit.sh
```

- Submit assign3.tar.gz to http://kayle.snu.ac.kr:37373/

```
> curl -F file=@assign3.tar.gz \
        -F key={your-apikey} \
    http://kayle.snu.ac.kr:37373/upload
```

Tips

- Please read the **malloclab-readme.pdf** carefully.

- If you have any questions, feel free to ask TAs via eTL.

Evaluation

- We provide 11 trace files for evaluation.
  - Total Score : 66 pts (6 pts / trace file)
  - Correctness (22 pts): You will receive full points if your solution passes the correctness tests.
  - Performance (44 pts): Space utilization + Throughput
    - To receive a good score, you must achieve a balance between utilization and throughput.

- You will get **zero points** if you break any of the rules or your code is buggy and crashes the driver.

- Due: 2022-10-28 (Friday) 23:59