jemalloc

Concurrent Programming

Introduction

- What is the jemalloc?
- How to use the jemalloc?
- Practice

What is the jemalloc?

- General purpose malloc(3) implementation that emphasizes **fragmentation avoidance** and **scalable concurrency** support

How to use the jemalloc?

- Install jemalloc

```
$ sudo apt-get install libjemalloc-dev
```

- Find path that jemalloc installed

```
$ sudo find / -name libjemalloc.so
```

How to use jemalloc?

- Append this line to ~/.bashrc
- Change path to your jemalloc installed path

```
export LD_PRELOAD="/path/to/your/libjemalloc.so"
```

```
112     . /etc/bash_completion
113   fi
114 fi
115
116 export LD_PRELOAD="/usr/lib/x86_64-linux-gnu/libjemalloc.so"
```

- Apply this change to the current session

```
$ source ~/.bashrc
```

Practice

- Allocate 40,000,000 memory blocks with 16bytes

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <pthread.h>
 4
 5 // allocate ~640MB
 6 #define MEMORY_BLOCK_SIZE   16
 7 #define NUM_BLOCK           40000000
 8
 9 int block_per_thread;
10
11 void* ThreadFunc(void* args) {
12     for (int i = 0; i < block_per_thread; i++) {
13         malloc(MEMORY_BLOCK_SIZE);
14     }
15
16     return NULL;
17 }
```

Practice

- Allocate 40,000,000 memory blocks with 16bytes

```
19 int main(int argc, char* argv[]) {
20     int num_thread = atoi(argv[1]);
21     block_per_thread = NUM_BLOCK / num_thread;
22
23     pthread_t* threads = (pthread_t*)malloc(sizeof(pthread_t*) * num_thread);
24
25     for (int i = 0; i < num_thread; i++) {
26         pthread_create(&threads[i], 0, ThreadFunc, NULL);
27     }
28     for (int i = 0; i < num_thread; i++) {
29         pthread_join(threads[i], NULL);
30     }
31
32     free(threads);
33
34     return 0;
35 }
```

Practice

- Without jemalloc

```
mrbin2002@ubuntu:~/TA/Multicore/lab12$ time ./prac_jemalloc 1

real    0m1.563s
user    0m1.172s
sys     0m0.388s
```

[allocate 16 block * 40,000,000 with 1 thread]

```
mrbin2002@ubuntu:~/TA/Multicore/lab12$ time ./prac_jemalloc 4

real    0m6.423s
user    0m3.120s
sys     0m10.640s
```

[allocate 16 block * 40,000,000 with 4 threads]

Practice

- With jemalloc

```
mrbin2002@ubuntu:~/TA/Multicore/lab12$ time ./prac_jemalloc 1

real    0m0.836s
user    0m0.744s
sys     0m0.088s
```

[allocate 16 block * 40,000,000 with 1 thread]

```
mrbin2002@ubuntu:~/TA/Multicore/lab12$ time ./prac_jemalloc 4

real    0m0.262s
user    0m0.876s
sys     0m0.080s
```

[allocate 16 block * 40,000,000 with 4 threads]

Thank You