Report

Malloc Assignment

Khushi Gauli

1002051688

I designed the benchmark test in such a way that it will test malloc algorithms on various criterias like runtime, splits, heap growth. The program that designed and coded for benchmarking various malloc algorithm is pasted below:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main()
{
clock_t start_time = clock();
int i =0;
while(i<1000000)
{
  char * ptr = ( char * ) malloc ( 650000 );
  free( ptr );
  char * ptr1 = ( char * ) malloc ( 100000 );
  char * ptr2 = ( char * ) malloc ( 300000 );
  char * ptr3 = ( char * ) malloc ( 200000 );
  free (ptr1);
  char *ptr5 = ( char * ) malloc ( 10000 );
  free (ptr2);
  free (ptr3);
  free (ptr5);
  i++;
}
 clock_t end_time = clock();
 double elapsed_time = ((double)(end_time - start_time)) / CLOCKS_PER_SEC;
 printf("Elapsed time: %.6f seconds\n", elapsed_time);



}
```

Test Result on basis of time (Run time is in seconds):

Algorithm Used	Sample1	Sample2	Sample3	Runtime(Avg)
System Malloc	0.159255	0.126463	0.154385	0.146367
FF Malloc	0.160049	0.168407	0.164027	0.164161
NF Malloc	4.039407	4.087009	3.934409	4.020275

BF Malloc	0.155573	0.146958	0.148866	0.150466
WF Malloc	0.159332	0.148174	0.147380	0.151629

EXPLANATION FOR THE TEST RESULT:
System Malloc has the best performance on the basis of time and Next Fit Malloc has the worst performance on the basis of time. I think Next Fit Malloc has the worst performance because it needs to keep track of another variable to continue from where it left.Other malloc algorithms do not need to check or keep track of other variable, it just starts from the beginning of the heap.

Test Results on the basis of grows and max heap:

Algorithm Used	Grows	Max Heap
FF Malloc	186	120900000
NF Malloc	1000001	1459939328
BF Malloc	187	121550000
WF Malloc	187	121550000

EXPLANATION FOR THE TEST RESULT:
In next fit allocation, the allocator starts searching for suitable memory blocks from the point where it left off in the previous allocation. If there's no space available, it continues from the beginning of the heap. This approach tends to result in fragmentation, but it allows the heap to expand more dynamically as it doesn't need to find the absolute best-fit block each time. It also grows the heap the most as we can see in in the result of the test.

Test Results on the basis of splits:

Algorithm Used	Splits
FF Malloc	4999692
NF Malloc	4000000
BF Malloc	4999814
WF Malloc	4999693

EXPLANATION FOR THE TEST RESULT:

Since the next fit allocator, grows the heap the most, it only makes sense for NF Malloc to have the least number of splits. Best Fit malloc has the most splits causing the most fragmentation among any other algorithms implemented.

Conclusion:

System malloc is best among all the malloc implementations on the basis of performance due to its long history of optimization. Among the implemented malloc algorithms, Next Fit is the worst one on the basis of runtime, heap growth but causes the least fragmentation (least splits). Other three algorithms are comparatively similar on the basis of run time but Best Fit causes the most fragmentation (highest splits).