

Profiling Lab

1 Objective:

This laboratory exercises guide you how to measure the program's performance using profiling. On completion of this laboratory, you should be able to

- Understand how to measure the program's running time
- Be aware the importance of tuning your program to be faster

2 Practice 1 - Profile

2.1 Activity 1 – familiarization with Profiling

Type the following program that is a simple one dimensional matrix. It will initialize it with a value of 1.0; it then forms two loops to initialize it.

```
void main() {  
    int i,j;  
    float a[10000];  
    for (i = 0; i <10000; i++)  
        a[i] =1.0;  
    for(i=0;j<10000;j++){  
        for (j=1;j<=10000;i++){  
            a[i] = a[j];  
        }  
    }  
}
```

2.2 Activity 2 – add function

Now type the program, you can open my file and use cut and paste. Please note that a subroutine called assign is created. Forget about the meaning of this program, it is designed to make you familiarize with the profiling only.

```
float a[10000];  
void assign(int i) {  
    int j;  
    for (j = 0; j <10000; j++)  
        a[i] = a[j];  
}  
void main() {  
    int i;  
    for (i = 0; i <10000; i++)  
        a[i] =1.0;  
    for (i = 0; i < 10000; i++)  
        assign(i);  
}
```

Now enable profiling, compile and run the above program. you will get the following screen.

C::B Profiler Results						
Gprof' s Output						
Flat Profile	Call Graph	Misc				
% time	cum. sec.	self sec.	calls	self ms/call	total ms/call	name
96.00	0.24	0.24	10000	24.00	24.00	assign
4.00	0.25	0.01				main

- 1) Explain why the hit count of assign() is 10000. (hint: look at the main())

- 2) How many functions in this program and explain why?

- 3) How many percentage of time the program spent in assign()?

- 4) The Total time is: 0.25 second in my machine (not the machine in the lab.). Please note that it varies form machine to machine. Explain the meaning of this 0.25 second.

2.3 Activity 3 – understands how the program affects the performance

Measure the total time of the following program with the same result.

Case 1

```
void main() {
    int i;
    int j;
    int max_num = 100000000;
    for (i = 0; i < max_num; i++)
        j = i/2;
}
```

The time is: _____

Case 2

```
void main() {
    unsigned int i;
    unsigned int j;
    unsigned int max_num = 100000000;
    for (i = 0; i < max_num; i++)
        j = i/2;
}
```

The time is:_____

Case 3

```
void main() {
    int i = 0;
    int j = 0;
    int max_num = 100000000;
    while (i < max_num){
        j = i/2;
        i = i + 1;
    }
}
```

The time is:_____

Case 4

```
void main() {
    int i = -1;
    int j = 0;
    int max_num = 100000000;
    while (++i < max_num){
        j = i/2;
        i = i + 1;
    }
}
```

The time is:_____

Case 5

```
void prt(int i) {
    int j = i/2;
}
void main(){
    int max_num = 100000000;
    for (int i = 0; i < max_num; i++)
        prt(i);
}
```

The time is:_____

3 Practice 2 - Determine your machine's performance

A program that can be used to determine MFLOP.

// This is matrix multiplication

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

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

```
void main(){
```

```
    float a[250][250], b[250][250], c[250][250];
```

```
    int i, j, k;
```

```
    for (i = 0; i < 250; i++)
```

```
        for (j = 0; j < 250; j++)
```

```
            for (k = 0; k < 250; k++)
```

```
                // matrix multiplication
```

```
                c[i][j] += a[i][k] * b[k][j];
```

```
}
```

3.1 Activity 1

Determine your machine's performance using timer functions provided by C library.

3.2 Activity 2

Determine your machine's performance using profiler