

LECTURE 7: PERFORMANCE MEASUREMENT

Computer Systems and Networks

Dr. Pallipuram

(vpallipuramkrishnamani@pacific.edu)

Lab Schedule

Today

- **Lab 5 – Performance Measurement is open**
- **Work on Lab 4**
- **Use Valgrind to resolve memory issues in Lab04?**

Lab 5 Goals

1. Measure program **execution time**
2. Break down program execution time by **specific subroutines / lines** of code
3. Monitor program for **memory leaks**
 - Not really “performance”, but uses same tool

Let's work with
memory leaks



Problem 1: Where is the memory leak in this code

```
int main() {
    int *array;
    array=(int *)malloc(sizeof(int)*1000);
    for(i=0;i<1000;i++)
        array[i] = i;
    return 0;
}
```

Problem 2: Where is the memory leak in this code

```
int main() {
    int **array;
    array=(int **)malloc(sizeof(int *)*1000);
    for(i=0;i<1000;i++)
        array[i] = (int *)malloc(sizeof(int)*500);
    free(array);
    return 0;
}
```

Problem 3: How do you fix the memory leak in this code

```
int main() {
    int **array;
    array=(int **)malloc(sizeof(int *)*1000);
    for(i=0;i<1000;i++)
        array[i] = (int *)malloc(sizeof(int)*500);
    free(array);
    return 0;
}
```

Problem 4: Find
leak in this code
and fix it




```
struct coordinate exit_queue(struct double_list **head) //
call by reference
{ struct coordinate seedpoint;
  struct double_list *temp;
  temp=*head;

  if(temp==NULL) //nothing on the list yet
  {
    printf("\n Nothing to exit..");
    exit(0);
  }
  else {
    seedpoint=temp->coord;
    (*head)=(*head)->next;

    if ((*head)!=NULL)
      (*head)->prev=NULL;
    return seedpoint;
  }
}
```

Valgrind



Valgrind Features

Memcheck module – Memory error detector

Access 1 beyond the beginning / end of your array?

Access un-initialized memory?

Reading/writing memory after calling free()?

Memory leak? (Lost pointer to a malloc() block)

Valgrind produces a report that identifies these errors

Valgrind Features

Callgrind module – Program profiler

Callgraph shows you what function called what other functions

How much CPU time does each function / code line consume?

Valgrind produces a report that summarizes CPU usage of your program



Valgrind Common Uses

Your program runs and suddenly segfaults

- Recall a segfault means a memory address was accessed that doesn't exist for your program

How do I find where this error is?

- Valgrind can monitor your program and detect accesses outside of static variables and dynamic memory regions



Valgrind Common Uses

Your program gets slower and slower the longer it runs

- Memory leak? (Slowing running out of heap memory because you malloc() without ever calling free())

How do I find where this error is?

- Valgrind can monitor your program. It can't tell you where you *should* free it, but it will tell you where you originally called malloc()



Valgrind Behind-the-Scenes

Just-in time compiler

- Your program is re-compiled onto a virtual (simulated) processor
- Another example of a virtual machine!

Benefit – Valgrind can observe your program running at the machine instruction level

Drawback – Slow! (5x slower than normal)

- *But it's still better than fixing bugs without a tool...*



Profiling Basics



Profiling Basics

The next labs (5-7) ask you to **measure application performance** by conducting experiments

- Execution time
- Processor usage
- Memory usage

Which of these system configuration do you think would be best in terms of producing the cleanest, most accurate, most reproducible results?



Terminal window titled "ECPE 170" showing a shell prompt and a large text overlay.

```
jshafer@ecpe170: ~  
jshafer@ecpe170:~$
```

Program to Benchmark

- DataNode modifications
- Good CV - Appli...le.pdf
- Trends Disk CPU N...k.xlsx
- Compile FreeBSD
- mount_nas
- Festival of Light_2008.ppt
- backup_mac_contacts



Terminal window showing a shell prompt:

```
jshafer@ecpe170: ~
jshafer@ecpe170:~$
```

Program to Benchmark

Downloads window showing a list of files:

Date	File Name	Size	Progress	Source
Today	ubuntu-12.04.1-desktop-i386.iso	476 KB/s - 36.2 MB of 695 MB, 23 mins left		http://releases.ubuntu.com/precise/ubuntu-12.04.1-desktop-i386.iso
Sep 18, 2012	05-c-programming.pdf	Removed		http://ecs-network.serv.pacific.edu/ecpe-170/slides/05cprogramming.pdf/at_do...
Sep 13, 2012	BIOS_Acor_7-10_A_A.zip	Removed		

iTunes window playing a song:

Alone in the Cosmos
Glint — Sound in Silence
0:19 / -4:22

Alone in the Cosmos
Glint
Sound in Silence

ESPN website screenshot showing sports scores and news:

ESPN: The Worldwide Leader

espn.go.com

EDITIONS: USA DEPORTES More

CITIES: BOSTON CHICAGO DALLAS LOS ANGELES NEW YORK

Bottom 9th	Top 8th	Top 7th	F/12	Final	Final	F/10	Final	Final
BAL 2 SEA 2	COL 0 SF 4	TEX 3 LAA 10	MIN 6 CLE 5	OAK 2 DET 12	MIL 6 PIT 0	ATL 3 MIA 4	BOS 7 TB 5	CIN 3 CHC 1

myESPN NFL MLB NBA NHL NCAA FB NCAA BB NASCAR SOCCER MORE SPORTS

TOP STORIES WATCH LIVE

HEADLINES MY HEADLINES SHARES

- Escobar banned 3 games for eye-black slur
- Falcons' Turner charged with DUI, speeding
- Ex-ref blasts Goodell | Sub refs under fire
- Late Ronaldo goal jolts City in thriller | Blog

Profiling Basics

The best approach (directly booting Linux) may not be convenient to achieve for this class

But you can ***definitely*** avoid the worst configuration!

- Keep your system **simple** when benchmarking