University of Arizona, Department of Computer Science

CSc 620 — Assignment 2 — Due midnight, Mon Sep 26 — 5%

Christian Collberg
September 12, 2005

1 Introduction

The purpose of this assignment is to learn x86 assembler, the PLTO binary editor, and the `ptrace` system call.

You can do this assignment in teams of two.

2 PLTO

Using PLTO, write a tool `ftrace` that annotates a program with calls to the library function `backtrace`, such that it prints out a stack trace whenever the function is entered. Your program should be called like this:

```
> gcc -g -static -Wl,-r ... -o program program.c
> ftrace program.r ... -o program.opt
> program.opt function-name
```

I.e., `ftrace` program should

1. read in the user's program using `plto`;
2. build the control-flow graphs for all the functions;
3. use `execinfo.h` to print out a backtrace whenever `function` is reached (see `http://www.delorie.com/gnu/docs/glibc/libc_665.html` and `http://www.helicontech.co.il/linuxprog.html` for more information);
4. write the program back out again.

3 ptrace

Write a tool `vtrace` that uses `ptrace` to set a breakpoint on a particular function and print out the current value of a particular global integer variable. Your program should be called like this:

```
> gcc -g ... -o program program.c
> vtrace program function variable
```

In other words, `vtrace` should

1. open the `program` executable file and, using the `libelf.h` library (see `http://developers.sun.com/solaris/articles/elf.html` and `http://www.linuxgazette.com/issue83/sandeep.html` for more information), and look up the addresses of `function` and `variable`;
2. start up the user's `program` using `ptrace`;
3. set a breakpoint at the beginning of `function`;
4. whenever the breakpoint is reached, get the value of `variable` (again, using `ptrace`) and print it out.

4 Submission and Assessment

The deadline for this assignment is midnight, Mon Sep 26. It is worth 5% of your final grade.

You should submit the assignmen electronically using the `Unix` command

```
turnin cs620.2 README ftrace.c vtrace.c README
```

.

`README` should briefly describe your implementation and list the members of your team.

Don't show your code to anyone, don't read anyone else's code, don't discuss the details of your code with anyone. If you need help with the assignment see the instructor.