



California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Spring 2022 – Lab 8 – *The Miner's Cats*

Overview

Working on the gold fields can be hard and lonely work – at times you have set out to find your own claim in the vast wilderness of the Sierra Nevada.

Unfortunately, even though you are the only human around for miles, you are never truly “alone”. Dozens of woodland creatures live all around you. This includes rats, mice, and other small creatures, that will happily steal your food.

It's a bit of a nuisance.

Fortunately, there is one woodland creature that became the friend of the Forty-Niners.

It looked very similar to the cat and soon became the de-facto pet of the miners. It helped kill rats and mice and, in return, got to live in the miners' campsites. And... most importantly... they were cute!

This creature was called, at the time, “The Miner's Cat”, but, nowadays, we call the “Ringtail”. The Miner Cat is closely related to racoons. They are incredibly easy to tame, incredibly smart, notorious trouble-makers, and an endless source of love and attention.

So, how do talk to your fluffy friend? What wonderful gift will it bring you?



Your Task

For this class, you are going to use two features of processors – the Vector Table and subroutines.

At the beginning of the semester, you wrote a lab that implemented the classic Hello World program. For the entire semester, you relied on the CSC35.o library. This hid the details of the operating system from you. Well, it's time to do the hard work – talk to the operating system directly.

In fact, you are not allowed to use the library!

To simulate your interaction with your new furry-friend, you are going to talk directly to the operating system.



Sample Run

The following is a sample run of the program. *You must change the wording of the text.* You don't need the blank lines – I added those for readability.

Oh! Hello little friend!

Put its name here.

You look rather cute today, _____!

Whoosh!

Sound effect of the Miner Cat leaving and coming.
Both were Printed by the subroutine

Whoosh!

Oh! Thank you! You brought me a _____.

So, your program will do the following:

1. Print a friendly message to your Miner Cat.
2. Say something nice to him/her (give them a nice name). Use its name in the blank.
3. Call a subroutine twice. It represents it jumping out of (and then into) your tent.
4. Finally, print some text telling everyone what the Miner Cat brought you. They are mischievous little thieves. In the blank above, put any fun item you can think of. You can change the structure of the sentence.
5. End the program



Talking to the Kernel

Let's start off by seeing if we output "Hello, World!" using UNIX Kernel calls. The basic calls, that you will need for this lab, are below.

Call	rax	rdi	rsi	rdx
Write	1	File Descriptor (1 = screen)	Source address	Total bytes to write.
Exit	60	Error Code (0 = all okay)	<i>none</i>	<i>none</i>

Tips

- You have to manually count the bytes for each string. The \0 was for the benefit of the PrintStringZ subroutine. It used the null character to count the bytes for you.
- Like all labs, **build it in pieces**. Get the "Exit" call to work first before working on the writes.
- You must setup all registers – **each time** – before you call the kernel.
- Pay close attention to the order of your instructions. Syscall **only** after you have all the registers ready.

Linking the Object File

Since you are **not** using the CSC 35 library. When you link, you will only specify the object you created.

```
ld -o a.out lab7.o
```

Requirements

You must think of a solution on your own. **Any lab using the csc35.o library will automatically receive a zero.** The requirements are as follows:

1. Print a greeting to the screen
2. Say hello to your furry-friend. Don't print _____. Create a name for your new friend.
3. Call the subroutine twice (since it leaves and comes back)
4. The subroutine should print some text.
5. Print some text letting us know what wonderful gift it brought you.
6. Exit your program

Submitting Your Lab

To submit your lab, you must run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

To submit your lab, send the assembly file (do not send the a.out or the object file to:

```
dcook@csus.edu
```



This activity may only be submitted in Intel Format.

Using AT&T format will result in a zero. Any work from a prior semester will receive a zero.



UNIX Commands

Editing

Action	Command	Notes
Edit File	<code>nano filename</code>	"Nano" is an easy to use text editor.
E-Mail	<code>alpine</code>	"Alpine" is text-based e-mail application. You will e-mail your assignments it.
Assemble File	<code>as -o object source</code>	Don't mix up the <i>objectfile</i> and <i>asmfile</i> fields. It will destroy your program!
Link File	<code>ld -o exe object(s)</code>	Link and create an executable file from one (or more) object files

Folder Navigation

Action	Command	Description
Change current folder	<code>cd foldername</code>	"Changes Directory"
Go to parent folder	<code>cd ..</code>	Think of it as the "back button".
Show current folder	<code>pwd</code>	Gives the current a file path
List files	<code>ls</code>	Lists the files in current directory.

File Organization

Action	Command	Description
Create folder	<code>mkdir foldername</code>	Folders are called directories in UNIX.
Copy file	<code>cp oldfile newfile</code>	Make a copy of an existing file
Move file	<code>mv filename foldername</code>	Moves a file to a destination folder
Rename file	<code>mv oldname newname</code>	Note: same command as "move".
Delete file	<code>rm filename</code>	Remove (delete) a file. There is no undo.