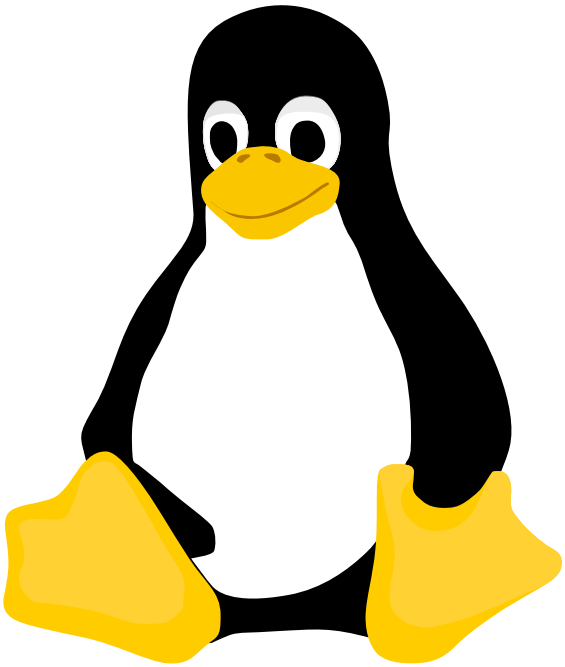
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Spring 2018 – Lab 8 – *Hello, Operating System*

Overview



At the beginning of the semester, you wrote a lab that implemented the classic Hello World program.

For the entire semester, you have been using the CSC35.o library. This library has hidden the details of the operating system from you. However, in this lab, you are finally going to talk directly to Linux.

Your challenge is to scan and print text to the screen (no numbers this time) using Linux kernal calls. It's just you and the operating system. So, you won't use the library this time.

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

Tips

- Like all labs, **build it in pieces**. First try to print the greeting to the screen and exit your program (2 calls)
- You have to setup all registers – **each time** – before you call the kernal.
- Pay close attention to the order of your instructions.

Requirements

You must think of a solution on your own. The requirements are as follows:

1. Prompt the user for their name.
2. Read the user's name from the keyboard.
3. Print "Hello " and the user's name. You can change the greeting to any text you want.
4. Give the user some advice on how to spend the Summer Break.
5. End the program
6. Use a separate kernal call for each requirement: 1 to 4. You will receive a zero if you don't.
7. Use the 64-bit system calls. You will receive a zero if you don't.

Sample Program

Here is a sample program. Your solution doesn't have to look exactly like this. The user's input is displayed in blue.

```
Please enter your name.

Morty

Hello Morty!
This summer, make sure to sleep in watch lots and lots of Netflix!
```

Kernal Calls

You will need to use the following kernal calls to complete this lab.

System Call	rax	rdi	rsi	rdx
Read	0	File Descriptor (0 = keyboard)	Target address	Maximum number of bytes
Write	1	File Descriptor (1 = screen)	Source address	Total number of bytes
Exit	60	Error Code *(0 = all okay)*	*none*	*none*

Submitting Your Lab

Run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

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

```
dcook@csus.edu
```

UNIX Commands

Editing

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

Folder Navigation

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

File Organization

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