



California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Fall 2017 – Lab 1 – *Hello World!*

Overview

It is a long standing tradition in computer science that your first program displays “Hello World” to the screen. This dates back to the first message sent over ARPANET – the predecessor to the Internet.

This week, you will basically get your feet wet with assembly programming. On the next page of this handout, there is a very basic “Hello World” program. Essentially, your only task for this lab is to print the traditional “Hello World!” to the screen followed by some other information.



Getting Started

1. Get an ECS Account. You probably already have one.
2. Run “putty” (Windows Secure Shell client) and login to either SP1, SP2, or SP3. Do not log in to Gaia or Athena.
3. I will walk you through how to copy the CSC35 Library into your new folder
4. You can use any text editor you want – such as **vim** or **nano**.

Creating a new file

To run nano, type “nano” following by the name of the file you either want to create or edit:

```
nano lab1.s
```

Hints

- First type in the program at the end of this handout verbatim. Then see if you can it to assemble, link, and execute. You don't have to type the comments, but I recommend it.
- Now work on each of the requirements below one at a time. You will turn in the final program, but incremental design is best for labs.

Hello World On x86 Linux

```
# lab1.s
# YOUR NAME HERE
#
# 1. Assemble : as -o lab1.o lab1.s
# 2. Link      : ld -o a.out lab1.o csc35.o
# 3. Execute   : a.out

.data                                #Start the data section
Message:                            #Message is an address
    .ascii "Hello, world!\n\0"      #Create a buffer of ASCII

.text                                #Start the text section
.global _start                      #Make the _start label public

_start:                             #UNIX starts here
    mov  $Message, %rax              #Put the address into rax
    call PrintCString               #Execute the csc35.o subroutine

    call EndProgram                 #Execute the csc35.o subroutine
```

Requirements

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

1. Print "hello world" to the screen. You can make it "howdy" or any type of greeting you like. This is pretty much done for you already in the sample code.
2. Print the text "My name is" and your full name to the screen.
Do **not** use the same string as in #1. Create a new label.
3. Print a quote from someone. It can be funny or inspirational.
Do **not** use the same string as in #1 or #2. Create a new label.
4. Print off the text "I will graduate in **year** from Sacramento State!" The year must be printed using the PrintInt subroutine.

Submitting Your Lab

To submit your lab, you must use Alpine – an easy-to-use application much like Nano. However, if your ECS account existed before this semester, you need to configure it to work on SP1, SP2, and SP3. To do this, please type the following verbatim and hit enter. You will only need to do this once.

```
cp /netdisk/skel/.pinerc ~
```

No space before the period. There is a space before the tilde.

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

alpine

Please send an e-mail to yourself (on your Outlook, Google account) to check if Alpine is working. 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	<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</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 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.