California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Summer 2019 – Lab 3 – *Vending Machine!*

Overview

Whether you are working a nine-to-five job or running from class to class, there is one machine that is in all our lives – the vending machine. These wonderful devices allow us to put in a few coins, select and item, and get what we want/need.

Vending machines vary from food to school supplies. So, the device is quite universal. For this lab, you get to create your own virtual vending machine.

Your program will display a menu of items, input the amount of cents, input their choice and, then, output their selection and their change.

Have fun!

You don't have to create a vending machine using stuff you find on campus. Use your imagination!

- Cat items
- Wizard items
- Cartoon items
- etc....

Example

Your solution doesn't have to look exactly like the example below. The user's input is printed in **blue**. The data outputted from your calculations is printed in **red**. You don't have to make the text that color in your program.

You don't have to worry about input validation. If they enter a wrong amount, don't worry about what will happen.

```
Fluff-a-matic Vending machine

1. Cat Nip (85 cents)
2. String (25 cents)
3. Bouncy Ball (60 cents)
4. Cat Snacks (42 cents)

Enter money: 100
Your selection: 3

3. Bouncy Ball (60 cents)
Your change is 40
```

Tips

The Data Section

- Create strings for each item in your vending machine.
- Create a table of those addresses (to lookup the purchased item).
- Also create a table of costs.
- **Work in your program in parts – incremental design!**

Example

The following contains an example of how to make an table of addresses (which are quads) and table of values (also quads). Please feel free to change your labels.

```
CatNip:
   .ascii "1. Cat Nip (85 cents)\n\0"

String:
   .ascii "2. String (25 cents)\n\0"

   …

Items:
    .quad CatNip
    .quad String
    …

Costs:
   .quad 85
   .quad 25
   …
```

Reading Integers

The CSC 35 Library has a subroutine called "ScanInt" that will read a value from the keyboard and store it into %rcx. This is equivalent to the Java Scanner class method "nextInt".

```
call ScanInt              #rcx = scanner.nextInt();
```

Requirements

1. Display a menu of items and costs. You must have (at least) three. Make sure to also print a name for your vending machine.

2. Input how much money they entered

3. Input their selection

4. Output the item they bought to the screen. You must use a table.

5. Output their change to the screen. You must use a table to look up the cost.

Submitting Your Lab

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 assembly 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** *object source*	Don't mix up the *objectfile* and *asmfile* fields. It will destroy your program!
Link File	**ld -o** *exe object(s)*	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.