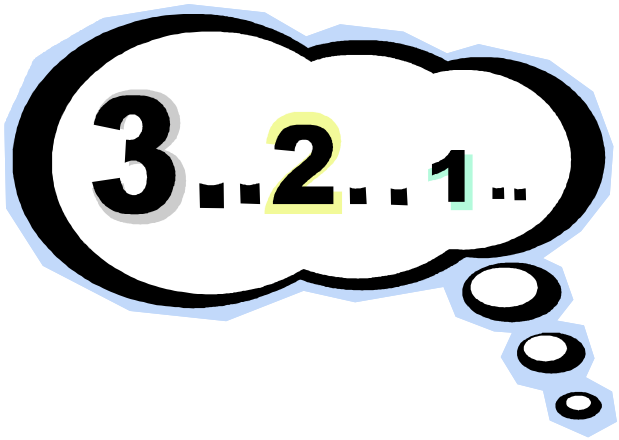
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Summer 2019 – Lab 5 – *Secret Number Game*

Overview



You have a wonderful job at a computer games company. Besides spending your days getting paid to play, you get to help write some simple games. For this lab, The Boss wants you to create a game based on the classic "Secret Number.

This game is also quite easy. The computer generates a random number (traditional from 1 to 100). Then the player will attempt to guess it. After each guess, the computer tells the player whether their guess is too high or too low. Once they get it correct, the game is complete.

Basically, you are going to write a loop that will continue until the guess is equal to the correct answer. Inside the loop, you need to check if the correct answer is too high or too low and display a message. The exact wording is up to you. Make sure to print a third message when the loop is complete.

The Game

The computer will generate a random number between 1 and 100. How do you do that? The csc35.o object library contains a subroutine called "random". Pass the range of numbers into %rax. It will return a random number from 1 to n-1 into %rax.

Please read the documentation!

Your solution doesn't to look exactly like the example below. But, make sure to fulfill all the requirements. You can use this output to test if your program is correct. The underlined text is user input.

```
Guess: 30
You are too low

Guess: 50
You are too high

Guess: 45
You are too high

Guess: 40
You are too low

Guess: 42
Correct!  It took you a total of 5 guesses!
```

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

1. Generate a random number
2. Loop until they enter the correct answer
3. Display a message if their guess is too high or too low in there.
4. Display a message congratulating the player when they get the correct answer.
5. Display the total number of player guesses

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.