Testing Program Implementation:

For this lab, we designed three interesting test programs, which are also utilizing our two new syscalls.

**Helper functions:**

1. **Random function**

This function will generate a random number between 0 and given parameter n. The logic is every time we call this function, it will use time() syscall to get current time and mode it by n.

In our programs, we used this function frequently, which really stress our kernel.

1. **Animation Print function**

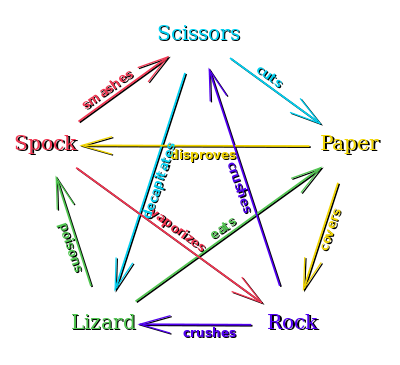
This function could dynamically display a string one letter by another. It can take in two parameters, one is the string for display, and the other is the time interval between each letter in millisecond.

This function calls sleep() syscall after displaying each letter.

**Program 1: rock scissor paper lizard and spock**

* Overview

This game is the updated version of rock scissor paper. Rules are in the below pictures.



* Implementation

Program will call the random function to get a random weapon and user can input a weapon too. It will display the result of the battle.

Program will exit if user presses any other keys besides 0-4.

**Program 2: recursive sleep**

* Overview

In this program, user can input two integers as the sleep time in each level and the number of levels. Program will call sleep in every level until it hit the bottom level.

* Implementation

Program used a recursive helper and called sleep() function in the helper function.

Program will exit naturally when it goes back to the original function.

**Program 3: Whack-A-Mole**

* Overview

In this program, it will display a 3\*3 sheet, with random moles in them. User can press q, w, e, a, s, d, z, x, c to hit according moles.

* Implementation

Program calls random function to get the random positions that moles will show up. Every time user presses a button (no matter hit the mole or not), if it has been 0.5 s since last mole is generated, it will generate another mole.

Program will exit if user press any other keys besides q, w, e, a, s, d, z, x, c.