Online on xv6 - System Call

Section: B1 Time: 30 minutes

Our favourite os xv6 doesn't have any built-in pseudo-random number generator. In this online, you will add a simple mechanism for generating random numbers and finally, add support for returning a *k*-length array of elements chosen from an *n*-length array..

Formally add 2 **system calls**.

1. setSeed(int z) : sets the seed for a pseudo-random number generator to z.
2. sample(struct array * arr, int k) : returns the k randomly selected numbers from the input array and updates the internal state. The structure *array* is defined as follows:

```
struct array{
      int len; // total length of array
      int array[15]; //array elements (input)
      int selected[15]; //randomly selected array elements
};
```

For ease of implementation, we will use a simple policy where calling *sample* will iteratively first increase the seed by 1 and the selected index will be *seed % array->len*. This will be done k times inside the *sample* function. The return value will be also a pointer to *struct array* object.

You also need to add 2 user commands.

1. seed n
2. sample k len [the_array elements]

See the sample I/O for clarification:

```
$ seed 2
The seed has been set to 2
$ sample 2 3 1 2 3
Sampled elements are [1, 2]
$ sample 3 5 10 20 30 40 50
Sampled elements are [50, 10, 20]
$ seed 12
The seed has been set to 12
$ sample 2 3 6 7 8
Sampled elements are [7, 8]
$ sample 3 1 5
```

```
Sampled elements are [5, 5, 5]
```

You can safely assume that the length of the array will be at most 15.

Submission:

```
git add --all
git diff HEAD > ../{studentID}.patch
```