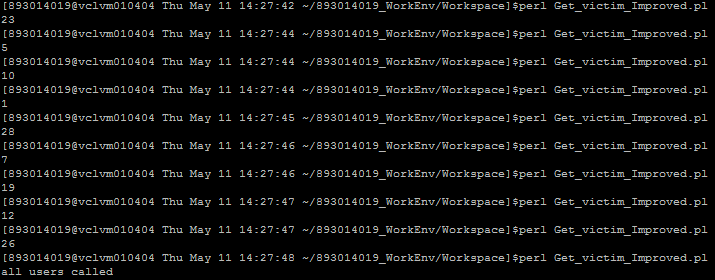
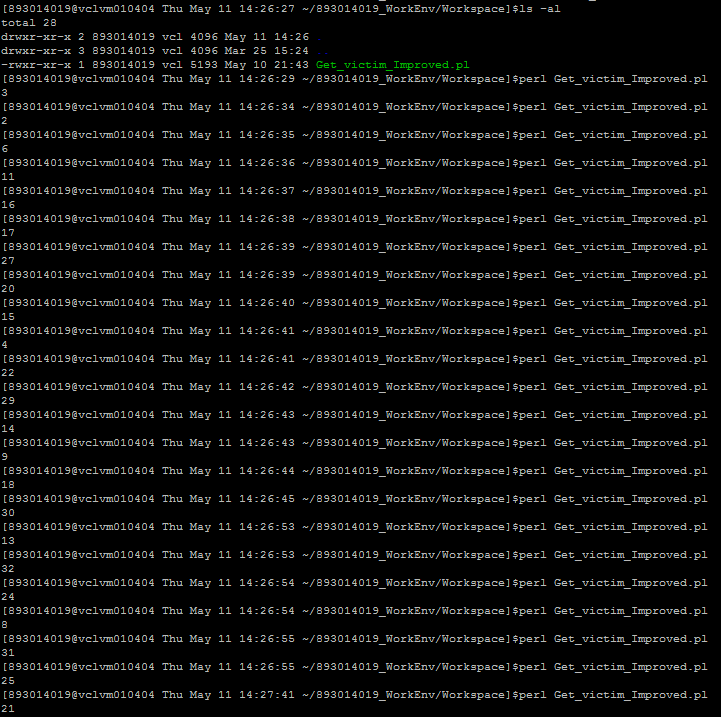
893014019

Andrew De La Fuente

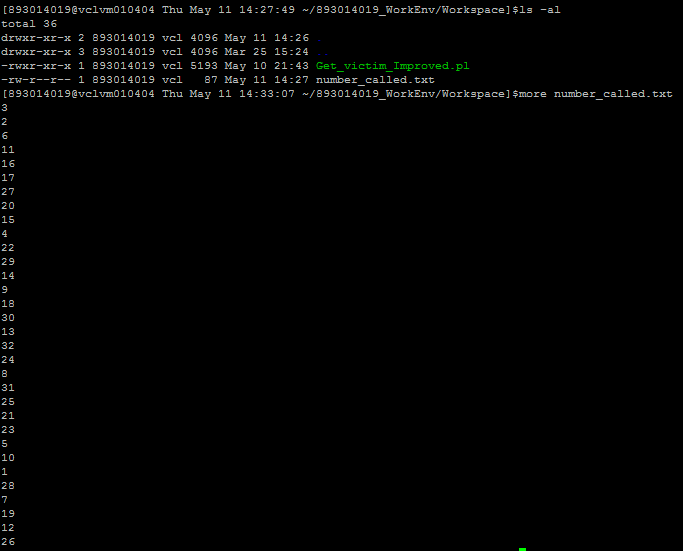
Get Victim Improved Assignment

This assignment is for **EXTRA CREDIT**.

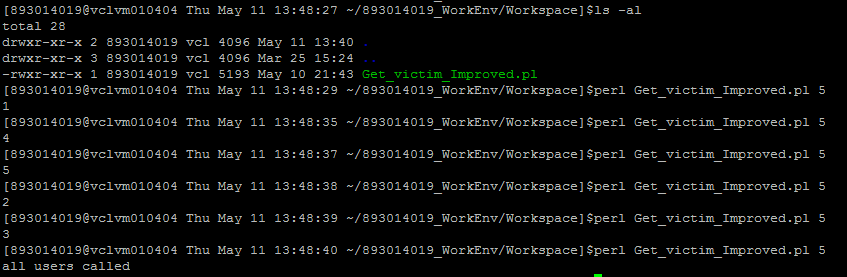
Default run of get\_Victim\_V1 – assumes 32 students



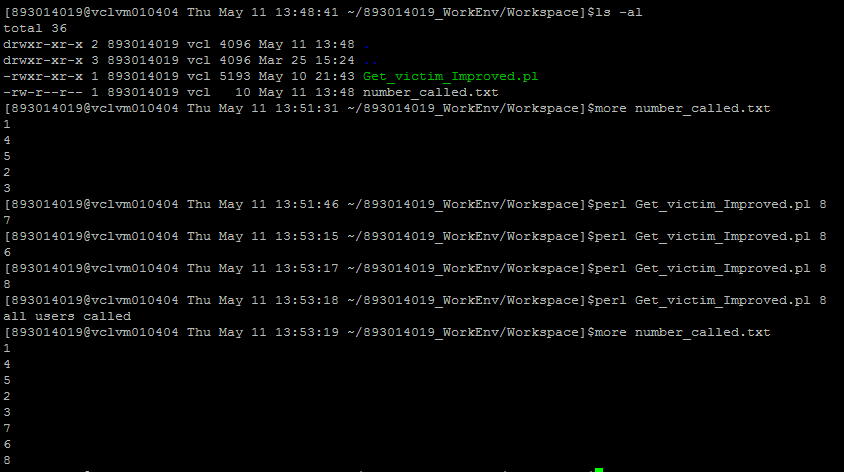
Print out number\_called.txt



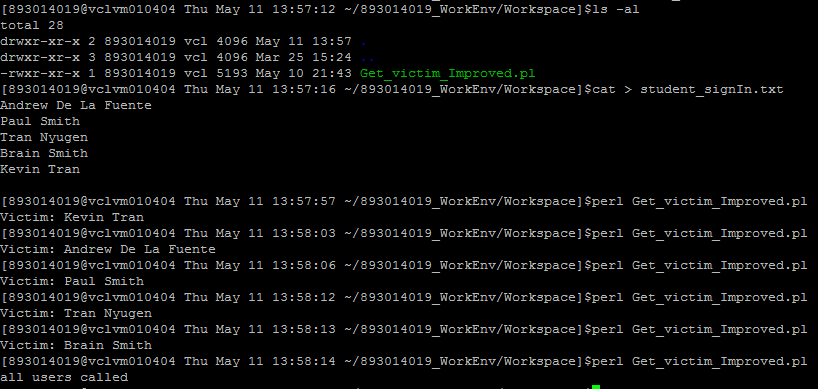
Running getVictim\_V1 functionality with 5 numbers and removed number\_called.txt



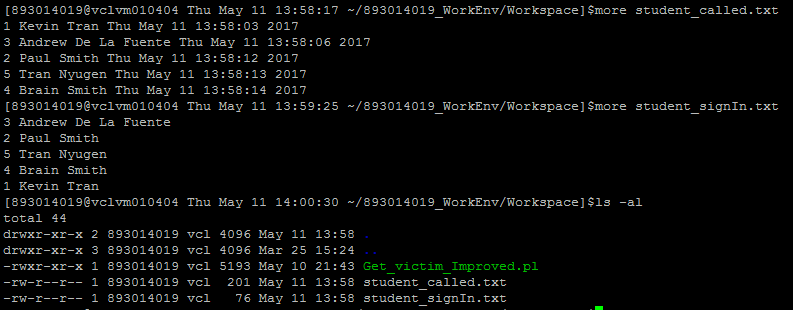
Print all numbers called and add 3 more students to the list



Running getVictim\_V2 with student sign in



Print out of student called archive file and the actual student signIn with pre-assigned random numbers



Code of Get\_victim.pl

#! /usr/bin/perl

use strict;

use warnings;

use Data::Dumper;

use Scalar::Util qw(looks\_like\_number);

my @number\_called;

my @student\_assign;

my @student\_called;

my @student\_List;

my $random\_number;

my $date = localtime();

if (-f "student\_signIn.txt") # use Get\_victim version 2.0

{

    open(my $read, '<', "student\_signIn.txt") or die "student\_signIn.txt NOT OPENED!!!!\n";

    while(my $row = <$read>)

    {

        chomp($row);

        push @student\_List, $row;

    }

    close($read);

    if (! -f "student\_called.txt" && !looks\_like\_number($student\_List[0])) # pre-assign values for all students and create "student\_called.txt"

    {

        my $count = 0;

        open(my $overwrite, '>', "student\_signIn.txt") or die "student\_signIn.txt NOT Wriiten Too!!!!\n";

        while (scalar(@student\_assign) < scalar(@student\_List))

        {

            my $bValid = 1;

            # if no matches found within the array

            while($bValid eq 1)

            {

                $bValid = 0;

                # Create a new random number if found on the list

                $random\_number = int(rand(scalar(@student\_List))) + 1;

                # search within the list

                for (my $i=0; $i < scalar(@student\_assign); $i++)

                {

                    my @number = split(" ",$student\_assign[$i]);

                    if (looks\_like\_number($number[0]))

                    {

                        if ($number[0] eq $random\_number)

                        {

                        $bValid = 1;

                        }

                    }

                }

            }

            $student\_assign[$count] = "$random\_number $student\_List[$count]";

            print $overwrite "$student\_assign[$count]\n";

            $count++;

        }

        close($overwrite);

        open(my $create, '>>', "student\_called.txt") or die "student\_called.txt NOT CREATED!!!!\n";

            open(my $read, '<', "student\_signIn.txt") or die "student\_signIn.txt NOT OPENED!!!!\n";

            while(my $row = <$read>)

            {

                chomp($row);

                my @line = split(" ",$row);

                if ( $line[0] eq $random\_number){

                    print $create "$row $date\n";

                    my @name = split(" ",$row);

                    print "Victim: ";

                    for (my $i=1; $i < scalar(@name); $i++)

                    {

                        print "$name[$i] ";

                    }

                    print "\n";

                }

            }

                close($read);

        close($create);

    }

    else # use pre-existing assigned values

    {

    open($read, '<', "student\_called.txt") or die "student\_called.txt NOT OPENED!!!!\n";

    while(my $row = <$read>)

    {

        chomp($row);

        push @student\_called, $row;

    }

    close($read);

    my $bValid = 1;

        if (scalar(@student\_called) < scalar(@student\_List))

        {

            my $student;

            # if no matches found within the array

            while($bValid eq 1)

            {

                $bValid = 0;

                # Create a new random number if found on the list

                $random\_number = int(rand(scalar(@student\_List))) + 1;

                # search within the list

                for (my $i=0; $i < scalar(@student\_called); $i++)

                {

                    my @number = split(" ",$student\_called[$i]);

                    if ($number[0] eq $random\_number)

                    {

                        $bValid = 1;

                    }

                    else

                    {

                        $student = $student\_List[$i];

                    }

                }

            }

            open(my $write, '>>', "student\_called.txt") or die "student\_called.txt NOT Wriiten Too!!!!\n";

            print $write "$student $date\n";

            my @name = split(" ",$student);

            print "Victim: ";

            for (my $i=1; $i < scalar(@name); $i++)

            {

                print "$name[$i] ";

            }

            print "\n";

            close($write);

        }

        else

        {

            print "all users called\n";

        }

    }

}

else # use Get\_victim version 1.0

{

    #gather all values that have been called

    if (-f "number\_called.txt"){

        open(my $fh, '<', "number\_called.txt") or die "number\_called.txt NOT OPENED!!!!\n";

        while(my $row = <$fh>)

        {

            chomp($row);

            push @number\_called, $row;

        }

        close($fh);

    }

    # Get random number

    if (@ARGV) #custom class size

    {

        my $bValid = 1;

        if (scalar(@number\_called) < $ARGV[0])

        {

            # if no matches found within the array

            while($bValid eq 1)

            {

                $bValid = 0;

                # Create a new random number if found on the list

                $random\_number = int(rand($ARGV[0])) + 1;

                # search within the list

                for (my $i=0; $i < scalar(@number\_called); $i++)

                {

                    if ($number\_called[$i] eq $random\_number)

                    {

                        $bValid = 1;

                    }

                }

            }

            open(my $fh, '>>', "number\_called.txt") or die "number\_called.txt NOT Wriiten Too!!!!\n";

            print $fh "$random\_number\n";

            print "$random\_number\n";

            close($fh);

        }

        else

        {

            print "all users called\n";

        }

    }

    else

    { #Default class size of 32

        my $bValid = 1;

        if (scalar(@number\_called) < 32)

        {

            # if no matches found within the array

            while($bValid eq 1)

            {

                $bValid = 0;

                # Create a new random number if found on the list

                $random\_number = int(rand(32)) + 1;

                # search within the list

                for (my $i=0; $i < scalar(@number\_called); $i++)

                {

                    if ($number\_called[$i] eq $random\_number)

                    {

                        $bValid = 1;

                    }

                }

            }

            open(my $fh, '>>', "number\_called.txt") or die "number\_called.txt NOT Wriiten Too!!!!\n";

            print $fh "$random\_number\n";

            print "$random\_number\n";

            close($fh);

        }

        else

        {

            print "all users called\n";

        }

    }

}