Creation and Inheritance of child processes with fork

11/20/2019

CECS 326

Dustin Martin - 015180085

dusmar6@gmail.com

Program desc:

This program demonstrates the capabilities of child processes, as well as how to terminate run-away child processes that have been stuck in an infinite loop.

The program prompts the user for a word-length, or to quit the program (-1). If the user does not enter -1, it will then spawn a child process to search through the string ‘doc’ for words of that length. If no words exist of that length (len<1 or len>13 for this given doc), then the child process will be stuck in an infinite loop due to the injected bug, prompting the user to kill the process manually.

Test Cases

	Test Case	Expected Result	Actual Result
1	User enters -1	Program exits	Program exits
2	User enters a length between 1 and 12 (lets say a length of **6** for this test case)	Program prints: “Words of length 6: 37”	Program prints: “Words of length 6: 37”
3	User enters a length which no word length in the document is equal to. (Lets say **0**)	Child process is caught in an infinite loop due to injected bug, continuously printing “.”	Child process is caught in an infinite loop due to injected bug, continuously printing “.”

Test case #3 notes:

To kill the child process without harming the parent process, run ‘ps aux’ in a separate terminal to get the child process’s PID (should be process just below the parent with the same name). Then type ‘kill-9 [pid] to kill the child process. You will then be able to continue entering word lengths to search for.

Source Code:

```
#include <iostream>
#include <cstdio>
#include <string>
#include <cctype>
#include <vector>
#include <sstream>
#include <unistd.h>
#include <cstdlib>

using namespace std;

string doc = "The primroses were over. Toward the edge of the wood,
where the ground became open and sloped down to an old fence and a
brambly ditch beyond, only a few fading patches of pale yellow still
showed among the dog's mercury and oak-tree roots. On the other side
of the fence, the upper part of the field was full of rabbit holes. In
places the grass was gone altogether and everywhere there were
clusters of dry droppings, through which nothing but the ragwort would
grow. A hundred yards away, at the bottom of the slope, ran the brook,
no more than three feet wide, half choked with king cups, watercress
and blue brook lime. The cart track crossed by a brick culvert and
climbed the opposite slope to a five- barred gate in the thorn hedge.
The gate led into the lane. The May sunset was red in clouds, and
there was still half an hour to twilight.The dry slope was dotted with
rabbits -- some nibbling at the thin grass near their holes, others
pushing further down to look for dandelions or perhaps a cowslip that
the rest had missed. Here and there one sat upright on an ant heap and
looked about, with ears erect and nose in the wind. But a blackbird,
singing undisturbed on the outskirts of the wood, showed that there
was nothing alarming there, and in the other direction, along the
brook, all was plain to be seen, empty and quiet. The warren was at
peace. At the top of the bank, close to the wild cherry where the
blackbird sang, was a little group of holes almost hidden by brambles.
In the green half-light, at the mouth of one of these holes, two
rabbits were sitting together side by side. At length, the larger of
the two came out, slipped along the bank under cover of the brambles
and so down into the ditch and up into the field. A few moments later
the other followed.";
```

```
int main(){

    istringstream stm(doc) ;
    string word ;
    vector<string> words;
    while( stm >> word ){
        words.push_back(word); //puts each word in a vector
    }

    int length;

    bool cont = true;
    while (cont){
        cout<<"Enter a length. Type -1 to exit."<<endl;
        cin >>length;
        cout<<"Initiating Search."<<endl;
        if (length == -1){
            exit(0);
        }
        int child = fork();
        if (child == 0){
            // in child

            bool found = false;
            int count = 0;
            while(!found){ //injected bug. If a word of that length is
not found (len <1 or len >12), then infinite loop
                cout<< "."<<endl;
                for ( const string& word: words){
                    if (word.size() == length){ //loops through
vector, checks length
                        count++;
                        found = true;
                    }
                }
            }
            cout<<"Words of length "<<length<<": "<<count<<endl;

            exit(0);
        }

    }
  return 0;
}
```