Ernie Argel

4/24/2019

CECS 326

[ernie.argel@student.csulb.edu](mailto:ernie.argel@student.csulb.edu)

The program allows the user to search a letter in array and retrieve the number of times it occurs in the array. First, the program asks the user to run the program or exit the program. If the user runs the program, it then asks them to enter the size of the array and the letter to search. Next, the program creates a child process of the parent with the fork() call. After, the child process executes after the line it was called and the parent process waits until it terminates/returns with the wait() call. Once, the child executes it searches through the array and counts the number of times it occurs in the array and outputs the information. If it doesn’t find the letter, it will just keep searching in the array and so the user must manually kill the process in another terminal. If it successfully finds an occurrence in the array it terminates and returns the execution back to the parent. The user is then asked if it wants to run the program again or exit.

**CPP CODE:**

//The software allows its user to input a letter to be searched in an array until the user chooses to terminate.

#include <unistd.h>

#include <iostream>

#include <cstdlib>

#include <ctype.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/wait.h>

using namespace std;

char alphabets[26] = {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

void search(int size, char c); //searches the letter through the array of size

int main() {

int size = 0; //size of array

char arr[size]; //array to search

char c; //letter to search

long childPID; //child process

while(true) { //allows user to keep running the program until they decide to terminate/exit

cout << "(1) Run Program \n";

cout << "(2) Exit Program \n";

string input;

cin >> input;

if(input == "1") {

cout << "Enter the size of array \n";

while(!(cin >> size)) { //integer validation for size of array

cout << "Enter the size of array \n";

cin.clear();

cin.ignore(123,'\n');

}

cout << "Enter letter to search \n";

cin >> c;

childPID = fork();

if(childPID > 0) { //parent process

cout << "Parent PID: " << getpid() << endl;

cout << "ppid: " << getppid() << endl;

cout << "child: " << childPID << endl;

wait(NULL);

} else if (childPID == 0) { //child process executes search method

cout << "childPID: " << getpid() << endl;

cout << "ppid: " << getppid() << endl;

cout << "child: " << childPID << endl;

search(size, toupper(c));

break;

}

} else if(input == "2") {

break;

}

}

exit(0);

}

//@param size of array is initialized

//@param c is the letter to be searched in the array

void search(int size, char c) {

char arr[size]; //array to search

int occurence = 0; //count for the number of times the letter occurs in the array

bool searching = true;

for(int j = 0; j < size; j++) { //initializes the array with random letters

arr[j] = alphabets[rand() % 26];

}

while(searching) { //keeps searching through the array until it kinds the letter (injected bug)

for(int i = 0; i < size; i++) { //iterates to the end of the array

if(arr[i] == c) { //if the letter is found increment the number of occurence by 1

occurence ++;

}

}

if(occurence > 0) { //if the number of occurence is greater than 0 then set searching to false and break out of the loop

searching = false;

}

}

cout << "Number of Occurence the letter " << c << ": " << occurence << endl; //outputs the number of occurence of the given letter in the array

}