Assignment 2

Hide Assignment Information

Instructions

Assignment overview

Large integer multiplication can be achieved by decomposing the two factors and performing multiplication and addition on the simplified terms. For example,

`1234*5678`

can also be written as

$[\ (12*56) * 10^4\ ] + [\ (12*78 + 34*56) * 10^2\ ] + [\ (34*78) * 10^0\ ]$

More generally,

1. assume n = length(a) = length(b), (can pad 0's for shorter number)
2. if length(a) <= 1 then return a * b
3. Partition a,b into a = a1 * $10^{n/2}$ + a2 and b = b1 * $10^{n/2}$ + b2
4. A = Multiply(a1, b1)
5. B = Multiply(a2, b1)
6. C = Multiply(a1, b2)
7. D = Multiply(a2, b2)
8. X = A $*$ 10^n
9. Y = (B + C) $*$ $10^{n/2}$
10. Z = D $*$ 10^0
11. Return X + Y + Z

For this assignment, write a C program which will:

1. Accept two 4-digit command-line parameters. (For simplicity, we will not worry about step 1 and 2 above - just deal with integers 1000-9999). If numbers are not in the correct range, be sure to fail gracefully.
2. Partition the two numbers in to a1, a2, b1, b2
3. Establish two pipes (Parent -> Child and Child -> Parent)
4. Fork a child process
    1. The parent will send a1 and b1 to the child through a pipe
    2. The child will multiply a1 and b1 and send A to the parent through a pipe. The parent will calculate X
    3. The parent will send a2 and b1 to the child through a pipe
    4. The child will multiply a2 and b1 and send B to the parent through a pipe
    5. The parent will send a1 and b2 to the child through a pipe
    6. The child will multiply a1 and b2 and send C to the parent through a pipe. The parent will calculate Y
    7. The parent will send a2 and b2 to the child through a pipe
    8. The child will multiply a2 and b2 and send D to the parent through a pipe. The parent will calculate Z
    9. (The child will exit)
5. The program will then calculate X + Y + Z and print the sum to the screen

Purpose

The goals of this assignment are the following:

- Learn how to use pipes for bi-directional communication between parent and child process.
- Get experience with the `pipe()` system function
- Gain more experience with the C programming language from an OS perspective

Computing platform

You are welcome to develop your program on your own workstation if you wish, but you are responsible for ensuring that your program compiles and runs without

error on the Gaul computing platform. Marks will be deducted if your program fails to compile, or your program runs into errors on Gaul.

- Gaul tips

Instructions

You will write a program called `assignment-2.c`. This program:

- Accepts two command-line parameters which are both 4-digit integers
- Prints out the communication between parent and child with identifying process ids
- Prints the product of the two integers using the method described above

Output

Executing `./assignment-2 1234 5678` should produce the following output:

```
Your integers are 1234 5678
Parent (PID 26580): created child (PID 26581)

###
# Calculating X
###
Parent (PID 26580): Sending 12 to child
Parent (PID 26580): Sending 56 to child
Child (PID 26581): Received 12 from parent
Child (PID 26581): Received 56 from parent
Child (PID 26581): Sending 672 to parent
Parent (PID 26580): Received 672 from child

###
# Calculating Y
###
Parent (PID 26580): Sending 12 to child
Parent (PID 26580): Sending 78 to child
Child (PID 26581): Received 12 from parent
Child (PID 26581): Received 78 from parent
Child (PID 26581): Sending 936 to parent
Parent (PID 26580): Received 936 from child
Parent (PID 26580): Sending 34 to child
```

```
Parent (PID 26580): Sending 56 to child
Child (PID 26581): Received 34 from parent
Child (PID 26581): Received 56 from parent
Child (PID 26581): Sending 1904 to parent
Parent (PID 26580): Received 1904 from child

###
# Calculating Z
###
Parent (PID 26580): Sending 34 to child
Parent (PID 26580): Sending 78 to child
Child (PID 26581): Received 34 from parent
Child (PID 26581): Received 78 from parent
Child (PID 26581): Sending 2652 to parent
Parent (PID 26580): Received 2652 from child

1234*5678 == 6720000 + 284000 + 2652 == 7006652
```

Helpful hints

- Since command-line parameters are read in as strings, you may want to take advantage of the `atoi()` function which will convert any string that looks like an integer into an integer. For example `atoi("1234")` will return the integer 1234.
- Decomposing a 4 digit number can be done as follows:

```c
int i = 1234;
int a = i%100;
int b = i/100;
```

Submitting

When you are finished your assignment, follow these steps:

1. Cite your work: Did you use code that you did not write yourself? Just like an essay, if you did not write it yourself, you must properly attribute the author. Leveraging existing code for your assignment is acceptable but it should represent a minor portion of your submission. Specifically, any code that fulfills the learning outcomes detailed in the **Purpose** section above

should be almost entirely your own. Contact the course instructor if you are unsure.
    1. If you used sample code from the course notes or the textbook, in the comments, state which sample program you used or which page(s) from the lecture notes you used.
    2. If you used code from an online source (eg. StackOverflow, GeeksForGeeks, etc.), in the comments, provide the link where the code was found.
    3. If you used an LLM such as ChatGPT, in the comments, provide the link to the prompts you used to generate the code.
    4. Using code from another student is a violation of the policy on academic integrity and is not permitted.
2. Verify your code runs on Gaul. Create screenshots demonstrating your program and place them in the `Assignment-2` directory. Remove any spurious files (eg. the executable version of your program). The directory should have a single file called `assignment-2.c` and any screenshots. Supporting files like `Makefiles` or `README`s are acceptable.
3. If necessary, run the following command to get out of the `Assignment-2` directory: `cd ..`
4. Package your assignment into a tarball: `tar -cvf Assignment-2.tar Assignment-2`
5. Verify the contents of your tarball (`tar -tvf Assignment-2.tar`) (`du -sh Assignment-2.tar`). The tarball should have your `assignment-2.c` file and all screenshots. If the `du` command says the tarball **is 10kb in size** you have an empty tarball and you made an error on this step. Make sure you are properly creating your tarball with the right files in it.
6. Use an SFTP program to download the tarball to your local workstation and then **upload it to OWL**. A blank submission or an invalid tarball cannot be graded.

Due on Feb 11, 2025 11:55 PM

Available on Jan 29, 2025 12:00 AM. **Access restricted before availability starts.**

Available until Feb 11, 2025 11:55 PM. **Access restricted after availability ends.**

Show Rubrics

Submit Assignment

Allowed File Extensions

tar

Files to submit

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File Record Audio Record Video

Comments

Submit Cancel