

Homework #11 CH-232

Arkid Kaleci

a.kaleci@jacobs-university.de

Problem 11.1:

- a) Assume the program has been compiled into `cnt` and that all system calls succeed at runtime. How many child processes are created for the following invocations of the program? Explain how you arrived at your answer.
- (1) `./cnt` (No child process is created; the `for` loop is never executed.)
 - (2) `./cnt 1` (One child process is created; the `for` loop is run once; the if body inside `action` which creates one child process is executed only once.)
 - (3) `./cnt 2` (Two child processes are created; the `for` loop is run once; the if body inside `action` which creates one child process is executed twice.)
 - (4) `./cnt 1 2 3` (Six child processes are created; the `for` loop is run three times; the if body inside `action` which creates one child process is executed once on the first run ($m = 1$), twice on the second run ($m = 2$), and three times on the third run ($m = 3$). Thus, $1 + 2 + 3 = 6$ total child processes.)

Explanation: this algorithm has a linear time complexity. Runtime is dictated by the number of arguments inputted in the initial program call. The `for` loop is called exactly once for each argument, and n child processes are created every call, where n is the argument. Thus, the time complexity is $T = \sum_{i=1}^{argc} a_i + 1$. The $(+1)$ in the end is for the parent process.

- b) Remove the line `exit(0)` and compile the program again. What is printed to the terminal and How many child processes are created for the following invocations of the program? Explain how you arrived at your answer.
- (1) `./cnt 1` (One child process is created; the `for` loop is run once; the if body inside `action` which creates one child process is executed only once. 2 lines printed)
 - (2) `./cnt 2` (Two child processes are created; the `for` loop is run once; the if body inside `action` which creates one child process is executed twice. 3 lines printed)
 - (3) `./cnt 1 2` (Three child processes are created; the `for` loop is run twice; the if body inside `action` which creates one child process is executed once on the first run ($m = 1$), and two times on the second run ($m = 2$). Thus, $1 + 2 = 3$ total child processes. 8 lines printed)
 - (4) `./cnt 1 2 3` (Six child processes are created; the `for` loop is run three times; the if body inside `action` which creates one child process is executed once on the first run ($m = 1$), two times on the second run ($m = 2$), and three times on the third run ($m = 3$). Thus, $1 + 2 + 3 = 6$ total child processes. 32 lines printed.)

What cause so many lines being inputted is the `stdio` being flushed twice when there is no `exit(0)`.

The amount of lines printed by the program can be found by the following equation:

$\# \text{ of lines printed} = \sum_{i=1}^{argc} (a_i + 1) * 2^{i-1} * \frac{(a_{i-1}+1)}{2}$. This equation works perfectly as long as there is no argument 0. Any argument 0 adds 1 to the number of printed lines and would be considered out of the sum series. (I derived this equation by many rounds of observations, multiple cases and corner cases.)

Problem 11.2:

a)

$\text{pow} :: \text{Integer} \rightarrow \text{Integer} \rightarrow \text{Integer}$

$\text{pow } x \ n$

| $n = 0$ $= 1$

| $n = 1$ $= x$

| $\text{even } (n)$ $= \text{pow } (x, n / 2) * \text{pow } (x, n / 2)$

| otherwise $= x * \text{pow } (x, n - 1)$