

# CS69003 Computing Systems Lab – I

## Autumn 2007

### Use of fork, exec and pipe

**Due: August 29, 2007**

#### **Part 1**

Write a multi-process program for the evaluation of simple arithmetic expressions. The parent process reads expressions from the user and creates a child process for the evaluation of each expression. For simplicity, restrict the operator to +, -, \*, /, % only, and assume that each operand is a positive integer. The expression is to be provided in the standard infix notation with one operator and two operands only (for example,  $2 + 3$ ,  $32 * 456$ , etc.).

The parent process runs a loop. During each iteration of the loop, the parent process creates a pipe for communication with a child. It then forks a new child process, reads an expression from the user, sends the expression to the child through the pipe, and waits for the child to terminate. If the child exits with a non-zero status, the parent process breaks the evaluation loop and terminates.

Each child process, in its turn, reads the expression from the pipe, parses it for obtaining the operator and the two operands, performs the desired operation, prints the result, and exits. If the operation was successfully performed by the child process, it exits with status 0, whereas if the evaluation of the expression fails, a non-zero exit status should be returned to the parent.

#### **Part 2**

Write another program that executes standard commands with zero or more pipe directives. The program gives a prompt and then waits for the user to enter a command. A command is read from the user and split based on the location(s) of the pipe character(s) appearing in the command. Each part of the command is then executed by a separate child process. In order to make the pipes in the command work, one needs to redirect `stdin` and `stdout` of the child processes to file descriptors created by the `pipe()` system call. The command `exit` terminates the program.

First, concentrate on commands without pipes and then on commands with only one pipe directive. For extra credit, modify your program to work for any number of pipes.

You can assume that the commands entered by the user are executable files in the current directory.

#### **Submission**

Write a program with the name `evalexp.c` for the first part of the assignment, and a program with the name `execcmd.c` for the second part. Mail both the programs together with the subject line:

`"Assignment 4 for 07CS60**"`.

## Sample output

```
[abhi@dedekind ass4]$ ./evalexp
[Enter expression]% 34 + 55
Child computes 34 + 55 = 89
[Enter expression]% 12345-45678
Child computes 12345 - 45678 = -33333
[Enter expression]% 16 * 35
Child computes 16 * 35 = 560
[Enter expression]% .
[abhi@dedekind ass4]$ ./execcmd
[Enter command]% ls -lp
total 52
-rw-r--r-- 1 abhi abhi 48 Aug 15 03:11 challenge.dat
-rw-r--r-- 1 abhi abhi 1074 Aug 14 23:01 child_nopipe.c
-rw-r--r-- 1 abhi abhi 2298 Aug 15 02:29 child_pipe.c
-rwxr-xr-x 1 abhi abhi 8518 Aug 16 16:21 evalexp
-rw-r--r-- 1 abhi abhi 3652 Aug 16 16:21 evalexp.c
-rwxr-xr-x 1 abhi abhi 9002 Aug 16 16:22 execcmd
-rw-r--r-- 1 abhi abhi 3617 Aug 16 16:21 execcmd.c
-rw-r--r-- 1 abhi abhi 1718 Aug 14 23:00 parent_nopipe.c
-rw-r--r-- 1 abhi abhi 3909 Aug 15 02:31 parent_pipe.c
[Enter command]% ls -lp | grep \.c
-rw-r--r-- 1 abhi abhi 1074 Aug 14 23:01 child_nopipe.c
-rw-r--r-- 1 abhi abhi 2298 Aug 15 02:29 child_pipe.c
-rw-r--r-- 1 abhi abhi 3652 Aug 16 16:21 evalexp.c
-rw-r--r-- 1 abhi abhi 3617 Aug 16 16:21 execcmd.c
-rw-r--r-- 1 abhi abhi 1718 Aug 14 23:00 parent_nopipe.c
-rw-r--r-- 1 abhi abhi 3909 Aug 15 02:31 parent_pipe.c
[Enter command]% ls -lp | grep \.c | sort
-rw-r--r-- 1 abhi abhi 1074 Aug 14 23:01 child_nopipe.c
-rw-r--r-- 1 abhi abhi 1718 Aug 14 23:00 parent_nopipe.c
-rw-r--r-- 1 abhi abhi 2298 Aug 15 02:29 child_pipe.c
-rw-r--r-- 1 abhi abhi 3617 Aug 16 16:21 execcmd.c
-rw-r--r-- 1 abhi abhi 3652 Aug 16 16:21 evalexp.c
-rw-r--r-- 1 abhi abhi 3909 Aug 15 02:31 parent_pipe.c
[Enter command]% ls -lp | grep \.c | sort | wc
      6      54     336
[Enter command]% ls -lp | grep \.c | sort | wc | wc -w
3
[Enter command]% exit
[abhi@dedekind ass4]$
```