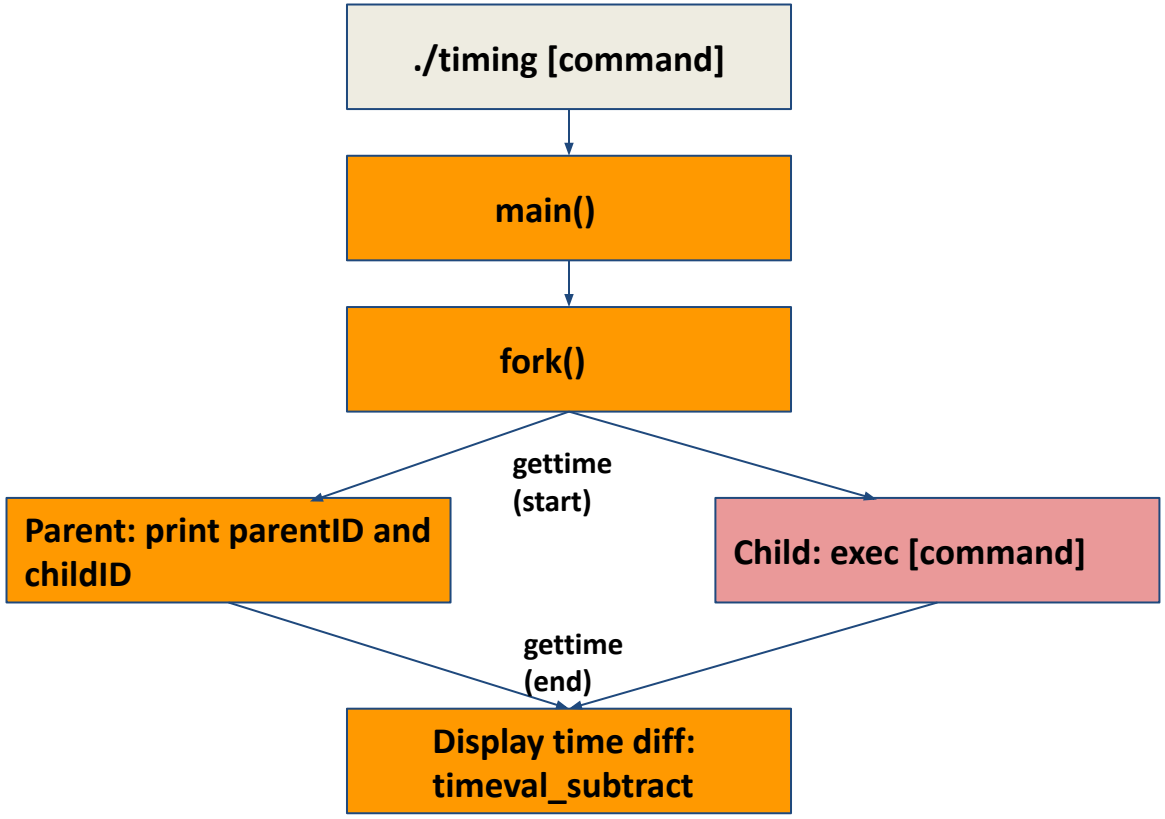
CS 350 Operating Systems
Spring 2024

Lab 3: Using fork, exec, and wait to time command execution

Task Flow


./timing [command]
main()
fork()
gettime
(start)
Parent: print parentID and childID
Child: exec [command]
gettime
(end)
Display time diff: timeval_subtract

What to do today

Write a program which does the following:

- Main process calls fork() to create child process.
- After fork() is successfully invoked, get a time stamp (start)
- In the parent, the parent prints:

  Parent (parent's pid) created child (child's pid)
- In the child, the child execute a program and its parameters which are provided by the user.
  - For example,
    - if user runs "./timing ls", the child execute command "ls"
    - If user runs "./timing ls –al", the child execute command "ls –al"
  - If the command does not exist, print

    Command "[the cmd user provided]" does not exist.
- After the above command is executed, get a time stamp (end)
- Get the difference between the two time stamps by calling:

  `timeval_subtract(&diff, &end, &start)`
- Report the run time of the command (i.e., `diff`). The print statement for the report has been given in the base code.

Getting a time stamp

- The gettimeofday() function can be used to get the current time.
  - Read the provided manual about the function to learn how to use it.
  - If timezone is not needed, NULL can be passed instead.

Learning the exec* family

- Read the provided PDF to learn the exec* family.
  - execl
  - execv
  - execlp
  - execvp
  - execle
- One of the exec functions is best suited for today's task
- Hint: the command and its params that the user wants to execute are passed in a vector.

Testing

- When you program is ready, perform the following test cases
  - ./timing ps
  - ./timing ls –al
  - ./timing sleep 1
  - ./timing sleep 2