

- 2) (4 points) Write a program in which the main routine creates a child process that prints the process ID of the parent. You may use `getpid()` but you may NOT use `getppid()`. You may not use IPC for this part (neither shared memory nor message passing).
- 3) (9 points) Write a program whose main routine obtains one parameter  $n$  from the user, i.e. passed to your program when it was invoked from the shell. Your program shall then create a shared memory and a child process.

The child process should obtain the value of  $n$  (you actually have multiple options for doing that) and create a sequence of length  $n$ , whose elements implement the equation  $z = (0.5y)^2$  where  $y$  is the index of the element, for example, the fourth element (index 3) shall have a value of 2.25. If  $n=4$ , then the sequence shall be 0, 0.25, 1, 2.25.

The child process shall create the elements, one at a time, and wait for a random interval of time (0 to 4.999 seconds) between generating elements of the sequence. As soon as an element is generated, the child places the element in the shared buffer by organizing it as described in slides 7-10 of lecture 5.

The parent process shall NOT wait for the child to exit but instead shall print an element as soon as it arrives into the shared buffer (again, in a manner similar to slides 7-10 of lecture 5)