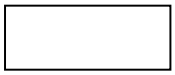
HW Chapter 3 **Name(s):_____**

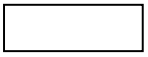
3.12 Including the initial parent process, how many processes are created by the program below?

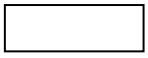
```
#include <stdio.h>
#include <unistd.h>

int main()
{
      int i;

      for (i=0; i<4; i++)
            fork();

      return 0;
}
```

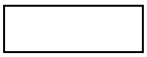


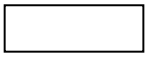
3.14 Using the program below, identify the values of `pid` and `pid1` that are output at lines A, B, C, and D. Assume that the actual pids of the parent and child are 2600 and 2603, respectively.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
      pid_t pid, pid1;

      pid = fork();
      if (pid < 0) {
            fprintf (stderr, "fork() failed\n");
            return(1);
      }
      else if (pid == 0)  {
            pid1 = getpid();
            printf ("pid = %d\n", pid);      // A
            printf ("pid1 = %d\n", pid1);    // B
      }
      else  {
            pid1 = getpid();
            printf ("pid = %d\n", pid);      // C
            printf ("pid1 = %d\n", pid1);    // D
            wait (NULL);
      }
      return 0;
}
```









Practical:

Find `_do_fork()`, the fundamental routine for creating a new process (i.e. the main fork-routine)

- What is the purpose (give a high-level description) of `copy_process()`?

  Straight from the Linux repository on Linus Torvald's GitHub: "This creates a new process as a copy of the old one, but does not actually start it yet. It copies the registers, and all the appropriate parts of the process environment (as per the clone flags). The actual kick-off is left to the caller.

- Within `copy_process`, what exact code guards against `fork()` bombs?

  if (nr_threads >= max_threads)
  goto bad_fork_cleanup_count;