Operating Systems Homework #3

Adam Razaq

1) a) Yes, because multiple processes can be created from the same executable file.

   b) No, because each process has its own address space.

2) a) A blocking system call.

   b) Hardware or timer interrupt.

3) a) No, because child processes are inherited by the init process.

   b) No, because its proccess ID becomes available for reuse, making it not become a zombie.

4) True

5) system, which executes a bash shell command
   strlen, which returns the length of a string

6) A blocking system call

7) The system call exit, to terminate the current process.

8) False

9) True

10) Voluntary context switch

11) False

12) False

13) False

13) The process is context-switched out and a separate kernel process starts execution.

14) a) The value of the variable a as printed in the child process is 5. when the parent value of a changes to 6, then it is scheduled next.

    b) Yes, since we are in child class. If we were in parent class, it would not succeed.

15) The value of count printed by the code above is 0.

16) With copy-on-write fork, the value of count printed by the code above is 0.

17) Whether the parent will block or not will depend on the system call variant and the options with which it is invoked.

18) fork-execwait

19) Ready

20) When P4 terminates, the process that must wait for and reap P4 should be process P3.

21) False

22) (M/K) * P bytes

23) False

24) True

25) The probability that the access results in a TLB hit and a subsequent page fault is (6/9)(1) = 2/3 = 0.66

26) $\frac{2^{33}}{2^{12}} = 2^{21}$

$2^{20} * (2^{21} + 10) = 4MB$

$\frac{2^{32}}{2^{12}} = 2^{20}$

$\frac{2^{42}}{2^{12}} = 2^{10} = 4GB + 4MB$

27) a) 0/16, 1/16, 20/16, 2/16, 20/16, 21/16, 32/16, 31/16, 0/16, 60/16, 0/16, 0/16, 16/16, 1/16, 17/16, 18/16, 32/16, 31/16, 0/16, 61/16

0, 0, 1, 0, 1, 1, 2, 1, 0, 3, 0, 0, 1, 0, 1, 1, 2, 1, 0, 3

b) 20, 2, 21, 32, 31, 60, 16, 17, 18, 61 10 virtual addresses using FIFO

c) LRU page replacement algorithm = 6

d) The lowest number of pages faults achievable would be 6 for both replacement optimal page algorithm and for optimal algorithm.

28)

20	40
10100	10 1000
00 11 0100	1011 1000
52	184