Homework 2 — due in class on Thursday 14 September

Total number of points available on this homework is 110. Full credit is equivalent to 100 points.

1. (20 pts) If we wanted to use a file to communicate between two processes, what issues would the programmer need explicitly to address?

2. (20 pts) What are the states that a process can be in and what causes it to change state?

3. (20 pts) With most operating systems, the PCB for a process is maintained in operating system memory, which can only be accessed and modified in kernel mode. Suppose the PCB for a process was maintained in the address space of the process and could therefore be accessed and modified by the process itself. What problems might this cause?

4. (20 pts) Compare the performance of the following program when it is compiled with the -DFAKESYSTEMCALLS=1 flag and when it is compiled with the -DFAKESYSTEMCALLS=0 flag. How long does it take to execute in each case? What causes the difference?

```
#include <stdlib.h>

#if FAKESYSTEMCALLS
int getpid() { return 55; }
#else
#include <unistd.h>
#endif

int main ( ) {
  int a;
  int limit = 10000000;
  int i;
  for (i = 0; i < limit; i++ ) {
    a = getpid ();
  }
}
```

5. (20 pts) Write a program under Unix with the following characteristics. A parent process creates two children. The parent then waits for the first child to complete, and kills the second when the first completes. After that, the parent exits.

6. (10 pts) In Unix, what is a zombie process?