CprE 308 Homework 1

Department of Electrical and Computer Engineering
Iowa State University

Student Name:

University-ID:

Problem 1 (5 points)
What is a process? What is a "Zombie" process?

Problem 2 (5 points)
Compare the advantage & disadvantage of Monolithic kernel and Microkernel.

Problem 3 (5 points)
What is the maximum depth of the stack for the following code in terms of number of stack frames (one stack frame per function call)? Be sure to count the first call to main.

```
int main() {
  f(12);
  return 1;
}

int f(int n) {
  if (n <= 0)
    return 0;
  else
    return f(n-1)+2*f(n-4);
}
```

Problem 4 (5 points)
(1) How many processes does the following code create? (2) Draw the process tree of the program.

```
int main() {
 int i;
 for (i=1; i<3; i++)
   fork();
 return 1;
}
```

Problem 5 (5 points)
Please write all possible outputs from the following piece of code:

```
int main(void) {
        pid_t pid = fork();
        if (pid > 0) {
                printf("I am the parent\n");
        } else if (pid == 0){
                printf("I am the child\n");
        }
        else printf("ERROR!\n");
        return 0;
}
```

Problem 6 (5 points)
Consider the following code. Assume all system calls return successfully and the actual process IDs of the parent and child during the execution are 2600 and 2603, respectively. What are the values of pid/pid1 at lines A, B, C, D?

```
int main() {
        pid_t pid, pid1;
        pid = fork();
        if (pid < 0) {
                fprintf(stderr, "Fork Failed");
                return 1;
        } else if (pid == 0){
                pid1 = getpid();
                printf("child: pid = %d", pid); /* A */
                printf("child: pid1 = %d", pid1); /* B */
        }
        } else {
                pid1 = getpid();
                printf("parent: pid = %d", pid); /* C */
                printf("parent: pid1 = %d", pid1); /* D */
                wait(NULL);
        }
        return 0;
}
```