Program 3

**Dued Date:** November 12, 2008

**Points:** 100

**Program**

1. *(100 points)* Modify the MINIX kernel to add a system call `get2pid` with the following interface:

   ```c
   int get2pid(pid_t *mypid, pid_t *parentpid)
   ```

   This system call returns the PID of the current process in the variable at the address contained in `mypid`, and the PID of the current process’ parent in the variable at the address contained in `parentpid`. If either argument is the `NULL` pointer, nothing for that particular variable is returned.

   The system call returns 0 if both pointers are valid for the calling program or `NULL`; otherwise it returns −1. Be sure to check that the addresses lie within user space for the calling process!

   **Hint:** To add the system call into the kernel, see how `getpid` and `getppid` work.

**Extra Credit**

1. *(80 points)* Modify the MINIX kernel to add a system call `getchpid` with the following interface:

   ```c
   int getchpid(int n, pid_t childpid[])
   ```

   This system call returns the PIDs of up to `n` children of the current process in the array `childpid` and returns the number of PIDs entered into the array. If either `n` is non-positive or `childpid` is `NULL`, the system call simply returns the number of child processes of the current process.

   If `childpid` points to an address outside the process’ user address space, or an error of any kind occurs, the system call returns −1.

   Write a program to demonstrate your call works.