These specifications describe the second subproject to complete phase two of Nachos. Continuing with the specifications from Nachos Project 2(a), here is the assignment:

1. The Exec() system call does not provide any way for the user program to pass parameters or arguments to the newly created address space. UNIX does allow this, for instance, to pass in command line arguments to the new address space. Implement this feature. The prototype for the new improved Exec is:

   SpaceId Exec(char* name, char* args[]);

Here, args is a vector of pointers to the strings which are the arguments passed into the child process. A NULL pointer terminates the vector. By convention, args[0] should be set to be the name of the program invoked (i.e., usually the last component in the absolute filename of the Exec-ed binary). When the Exec-ed user process begins execution, the main function is essentially called as

   main(int argc, char* argv[])

The integer argc is the argument count. Since argv[0] is always set by convention, argc should be at least 1. The last program argument will be denoted by argv[argc-1].

2. Write a shell and some utility programs. A shell reads a command from the user via the console, then runs the command by invoking the kernel system call Exec(). The UNIX program bash is an example of a shell. There is already a simple shell in the test directory. Enhance it to do some interesting things. Also adapt the shell to invoke programs with command line arguments (using the Exec() you wrote above). Test your shell and system call handling by writing a couple of utility programs, such as UNIX cat, cp, etc.

---

1 From the original Nachos Assignment #2 by Tom Anderson; modified by Phil Kearns and later Joel Hollingsworth, Barry Lawson and Lewis Barnett