Adding shell commands.

You will adding all the commands as shell programs, we have described below the steps for adding a shell command. This is adapted from exinu wiki; see http://xinu.mscs.mu.edu/Shell This page was added to the wiki on our request.

Lets say we want to add an echo command that echo s the words that list after the command "echo"

1. Update include directory shell.h file include/shell.h to include prototype

```
command xsh_echo(ushort stdin, ushort stdout, ushort stderr, ushort nargs,
char *args[])
```

2. Update shell/shell.c command table to include command as an entry

```
{echo, FALSE, xsh_echo}
```

```
The second parameter says "FALSE" because echo is not a built-in
function.
```

3. Add xsh_command.c in the shell directory

Add xsh_echo.c file in the shell directory.
Its content is as follows:

```
=======================================================================
====
/**
 * @file      xsh_echo.c
 * @provides xsh_echo
 *
 * $Id$
 */
/* Embedded XINU, Copyright (C) 2007.  All rights reserved. */

#include <kernel.h>
#include <stdio.h>
#include <string.h>

/**
 * Shell command echos input text to standard out.
 * @param stdin descriptor of input device
 * @param stdout descriptor of output device
 * @param stderr descriptor of error device
 * @param args array of arguments
 * @return OK for success, SYSERR for syntax error
 */
command xsh_echo(ushort stdout, ushort stdin, ushort stderr, ushort nargs,
char *args[])
{
    int i;  /* counter for looping through arguments */

    /* Output help, if '--help' argument was supplied */
    if (nargs == 2 && strncmp(args[1],"--help",6) == 0)
    {
        fprintf(stdout, "Usage: clear\n");
        fprintf(stdout, "Clears the terminal.\n");
```

```
            fprintf(stdout, "\t--help\t display this help and exit\n");
            return SYSERR;
    }

    /* loop through the arguments printing each as it is displayed */
    for ( i = 1; i < nargs; i++ )
    {
        fprintf(stdout, "%s ", args[i]);
    }

    /* Just so the next prompt doesn't run on to this line */
    fprintf(stdout, "\n");

    /* there were no errors so, return OK */
    return OK;
}
```
==

4. Update compile/Makefile to add the newly created shell file to be included in the compilation process.
   In the SHL=…
   add xsh_echo.c
5. Compile and upload
   cd compile
   make
   xinu-status
   (find a free server: do not use the server name with a 1 at the end,. Example brylow1)

   To upload:
   mips-console server_name
   ex: mips-cosole moiz

6. If everything worked alright you will see xsh> prompt.
7. Type help to see "echo" listed as one of the shell commands
8. Type the command "echo" and some words. You will see the words you types echoed back.