

## Project B

*Due: May 3, 2006*

### Overview

You are to extend your Project A to remove some restrictions and add new features. The command has the following format:

```
xssh [-x] [File [Arg] ... ]
```

The difference here is that the optional `File` argument indicates an `xssh` script. The `File` argument can be followed by zero or more arguments that are assigned to the variables `$1`, `$2`, etc., just as in normal shells. Note that `File` is assigned to the variable `$0`.

Here are the changes to existing features and commands:

- `xssh` should continue to run even in the face of errors.
- All builtin commands should be checked for the proper number of arguments if possible.

This version of `xssh` should support pipelines, and the extended background command. It should also respond to the terminal interrupt character (normally `ctrl-c`) as described below. Here are some specifics:

- A command has the following syntax: `[ < InFile ] [ > OutFile ] Command` where `Command` can include a pipeline. Note that white space separates the symbols.
- A pipeline in the foreground has the same syntax as in other shells; i.e., `'Command | Command | ... | Command'` is a pipeline. Note that the `Commands` can have arguments. Although there are a variety of ways to implement a pipeline, full credit will only be given if `xssh` is the parent of all processes in the pipeline.
- A pipeline in the background is expressed as `'bg Command | ... | Command'`; i.e., it combines our earlier `bg` command with the pipeline symbol `'|'`.
- `stdin` and `stdout` redirection is indicated by preceding a command with the following optional sequence: `[ < InFile ] [ > OutFile ]`.
- The terminal interrupt character (normally `ctrl-c`) generates the interrupt signal (`SIGINT`). `xssh` should terminate the foreground process(es) and return to the command line prompt. However, it should not terminate background processes or `xssh`.
- The user should be notified about any background jobs that have terminated right after the command line prompt is displayed.
- The extended background command allows `xssh` to control execution of backgrounded processes in a limited fashion. The command syntax is identical to the background command except that it is indicated by the `+bg` command (note the extra `+` character). The difference is that all processes run in as an extended background process are scheduled by `xssh` so that they get an approximate equal share of the CPU.

## What to Submit

The CS422S Web page contains a link to the documentation template. You should complete the template and submit it in both hardcopy AND electronic form. Submit the completed documentation template AND a listing of the source code. The electronic submission (described below) should include the completed documentation template, the source code, the Makefile, test scripts, and test output. The electronic copy is due by midnight May 3. The hardcopy can be submitted to my office by noon the next day (May 4).

**Early Submission:** You can submit a shortened documentation template if you can submit **BEFORE** May 3. Here are the requirements for early submission:

- The program must be bug free (as far as you can tell).
- Schedule a demonstration of your program with me **BEFORE** May 2. You will be asked to demonstrate your program, and you may be quized about aspects of the program.
- There is no hardcopy submission.
- The shortened documentation template contains only instructions for running and testing your project and a full description of any extra credit work.
- Submit electronically **BEFORE** May 3. The README file contains the shortened documentation template instead of the full template.

## Electronic Submission

The end result should be that you mail to kenw@arl.wustl.edu a single *shar* (shell archive) file containing your files. Do **NOT** submit object code or executables. The following commands will create a shar file named B.shar containing the files xssh.c, in1.txt, out1.txt, and other files and then send mail to me:

```
shar README Makefile xssh.c ... in1.txt out1.txt ... > B.shar
mail -s B.shar kenw@arl.wustl.edu < B.shar      # mail is usually in /bin
```

The README file is the completed documentation template.

## Late Policy

Late submissions will be accepted in some cases but must be negotiated prior to May 3. The standard penalty is 20%, and May graduates may limited in their options.

## Extra Credit

Here are some possible extra credit topics (you are allowed to pick one):

- Job Control (10 Points): Respond to ctrl-z in a standard fashion, and provide support for the `bg`, `fg` and `jobs` commands.
- Torous Communication (10 Points): Provide support for the `torous` command. Its syntax is `torous N <n1> Command 1 <n1> ... <n1> Command N <n1>` where `Command` could be a pipeline and `<n1>` is the newline character. The command sets up a torous communication structure (using named pipes) between the `N` commands where `N` is the square of an integer. The processes are run in the background, and `xssh` regulates fairness much like the `+bg` command. See me if you are interested in this problem.