

Operating Systems, Program 1

Daniel Nix

February 6, 2015

Program Description The D-shell is a simple shell to provide the user with information about processes running on the system. This includes providing the name of a process given a specific process ID (pid), providing the pids of all programs containing a user provided command string, and print system information. This is done with three commands:

- `cmdnm <pid>`
- `pid <command>`
- `systat`

1 Algorithm

1.1 Event Loop

The event loop is handled in main. It is a simple while loop that checks if the most recent command was “exit”. As long as “exit” has not been executed, the program continues to accept commands from the user.

Each command entered is compared to one of the accepted commands. If the command is valid, the appropriate function call is made. Otherwise, no functions are called. Either way, the function to print the appropriate commands is called and it determines whether the given command valid or if it was a blank line.

1.2 Parsing

Command parsing is done in the “parse_first_arg” function. It strips the command name and any additional arguments so that the command may run even if extra command line parameters are entered. It also prints a usage message to the

user if there was no argument provided when there should have been.

1.3 cmdnm

The command “cmdnm” is responsible for printing the name of a process with a given pid. It accepts one parameter that is the pid of the process name to be found. Parse_first_arg is called to get the first argument. If none is provided or if the argument is non-numeric, the program notifies the user and does not proceed to find an invalid pid.

If a numeric pid is provided the program attempts to open the `/proc/<pid>/cmdline` file. If the directory does not exist then the open will fail and the user is notified that the process is running. Otherwise the `cmdline` file is read and the command string is output to the user.

1.4 pid

The command “pid” is responsible for accepting a string from the user as an argument and printing the pid of all programs with the argument as a substring.

The user’s command is provided to the function and parse_first_arg is called to extract the first argument. If no argument is provided the function exits.

The function then uses the `dirent.h` library to produce a list of all numeric directories in `/proc`. This list is all pids currently in use. The list is then looped over, each process’s `/proc/<pid>/cmdline` file is read, and an attempt is made to find the user’s argument as a substring. If it is found then the current directory name, which is the pid, is output along with

the command string.

1.5 sysstat

The command “sysstat” prints out system information to the user. The data is found in the /proc directories in the files:

- version
- uptime
- meminfo
- cpuinfo

Linux version, system uptime, memory information, and cpu information are found in these files and output to the terminal.

2 Testing

The D-shell was first tested for correctness. The commands were run correctly and output was correct when compared to the system command ps -aux. Blank lines with varying numbers of spaces and tabs were also tested.

Incorrect cases were also tested. These included:

- commands with missing parameters (print usage message)
- commands with too many parameters (should execute as normal, using the first parameter as expected, ignoring the rest)
- commands such as “cmdnms” (print correct command list)

Once all cases had passed the program was considered correct.

3 Submission Description

3.1 Compiling Instructions

To compile the source code, unzip the prog1.tar file. From the prog1 directory, type “make” and the source code will be compiled. The dash executable will be placed in the root prog1 directory. Run the D-Shell with the command “./dash”. All output is to the terminal and the only external files are in the /proc directory, created by the kernel.

3.2 External Functions

No external functions were required to complete the D-shell