UCE Birmingham

technology innovation centre

tic

In-Course Assessment Brief

Undergraduate Programme Academic Year 2004/2005

Module:	**Programming for Communications and Networks V**
Semester:	1
Assignment 2:	Coursework
Division:	Electronics, Communications and Software
Module Co-ordinator:	Richard Kay
Set Date:	Thurs 7 Oct 2004
Hand-in Date:	Writeups: Thurs 3 Feb 2005. See assignment details for demonstration dates for individual program demonstrations.
Hand-in Method:	Submitted through the box system, to the module co-ordinator.
Nominal time to complete this assignment:	40 Hours
Brief Assessment Details **Assessment Weighting:**	This assessment comprises 4 small programs to be designed, implemented, tested, demonstrated and documented by students. This assignment is worth 50% of the total module mark.
Individual Assessment:	Individual assessment. The work you submit shall be your own and not the product of collaboration with anyone else. Plagiarism will be penalised. Group assessments. Members of the group shall be listed in the assessment.

Learning Outcomes to be Assessed On completing this assignment students will be able to write programs which use various means of communicating with other programs.
Assessment Details: See below.
Assessment Criteria: Program 1 is worth 40%. Programs 2,3 and 4 are worth 20% each. Within each program, marks are divided equally between (a.) demonstration, (b.) source code, (c.) documentation and (d.) quality and completeness.

Students are required to design, implement, test, demonstrate and document the following programs. Documentation to include specification, a test plan and a 200 word review of what was learned for each program. Feedback sheets resulting from demonstrations must be included in final coursework submitted.

Program 1. 40% demonstration due week 8.

Students must write a script which analyses a mail log file created by the Sendmail email relay program. The language recommended for this application is Bash shell script, but other scripting languages available on the server (e.g. Other Unix shells, Perl, Python) may also be used. The mail log file will be made available before week 3 through the module web site. Students are expected to automate use of utility programs such as awk, sed, grep, sort etc, chaining use of these programs through use of pipelines and temporary files and I/O redirection facilities as provided by the bash shell script language, or use comparable facilities in the scripting language chosen.

The output resulting from this automated analysis is required to list the incoming IP addresses from which mail messages are received or rejected. The output list should be in descending order of number of attempts to send messages from each IP address, and within this order the output list should be sorted by IP address in ascending order. Log lines concerning incoming messages all contain the string: "from" and an externally routable IP address in dot quad format (e.g. 11.254.35.27). Email addresses within the log file supplied have been changed for reasons of privacy.

Demonstration of programs 2, 3 and 4.

Note that while students are required to be in a position to demonstrate any or all of programs 2, 3 and 4 by week 13, the time available for demonstrations is limited in practice. Staff will therefore decide which of these programs must be demonstrated based on the time available, and demonstration times in addition to timetabled tutorials may optionally be booked if required. Communications concerning booking of specific demonstration times require that students read announcements made on the pcn5 email list referenced on the module website.

Program 2. 20% demonstration due week 13

Process creation. The creation of a child process that runs another program, specifiable by the user. The process-IDs and parent-process-IDs of the parent and child should be displayed, along with informative text on which process is currently executing. Include options for specifying whether or not the parent should wait for the child to terminate.

Program 3. 20% demonstration due week 13

The use of signals. Write a program that contains functions to handle SIGINT and SIGQUIT signals. The functions should print out appropriate messages to indicate which signal has been detected . They should also ignore both SIGINT and SIGQUIT whilst they are handling the signal. (Explain why this should be so).

Program 4. 20% demonstration due week 13

Interprocess communication – pipes. Write a program that creates a pipe and forks two children. Each child then writes to the pipe. The parent reads from the pipe and prints out any data that it receives. Write another program that uses two pipes operating in different directions to give two-way communication between two processes.