Linux C Programming

SCHOOL OF ADVANCED TECHNOLOGY

Course Number: CST8204	**Contribution to Program:** Vocational Core	**Normative Hours:** 75
Applicable Program(s):	**AAL:**	**Approval Date:** 01/01/0001
0006X01FWO Computer Eng. Technology - Comp. Science	2	
0006X03FWO Computer Eng. Technology - Comp. Science	2	
Prepared by:		**Approved by:**
Co-Requisites N/A		**Approved for Academic Year:**
Pre-Requisites CST8110 and CST8201		

COURSE DESCRIPTION

This course introduces students to the basics of the C programming language, Unix/Linux Shell Scripting, and builds upon the foundation laid for systematic problem solving and the basics of the Unix/Linux operating systems. This course addresses application design, development, debugging, and testing in the UNIX/Linux operating system environment. It covers in some detail the essential elements of shell scripting in bash, and also prepares the student for the successor courses by addressing regular expressions. System functions will be used in the C programming language for handling I/O (both buffered and unbuffered) and file system resources, processes, and memory.

RELATIONSHIP TO VOCATIONAL LEARNING OUTCOMES

This course contributes to your program by helping you achieve the following Vocational Learning Outcomes:

Computer Eng. Technology - Comp. Science 0006X01FWO

1	Diagnose, solve, troubleshoot, and document technical problems involving computing devices using appropriate methodologies. (T,A)
6	Analyze, build, test, implement, and maintain applications. (T,A)
8	Articulate, defend, and conform to workplace expectations found in technology environments. (T,A)

Computer Eng. Technology - Comp. Science 0006X03FWO

ESSENTIAL EMPLOYABILITY SKILLS

The course contributes to your program by helping you achieve the following Essential Employability Skills:

1	Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience.(T,A)
4	Apply a systematic approach to solve problems.(T,A)
5	Use a variety of thinking skills to anticipate and solve problems.(T,A)
10	Manage the use of time and other resources to complete projects.(T,A)

COURSE LEARNING REQUIREMENTS/EMBEDDED KNOWLEDGE AND SKILLS

COURSE LEARNING REQUIREMENTS **When you have earned credit for this course, you will have demonstrated the ability to:**	**EMBEDDED KNOWLEDGE AND SKILLS**
1. Develop complete programs using top-down design and structured programming techniques	• Understand and use strict ANSI C (C89; also known as ISO C90) without compiler or library extensions, complete with suitable documentation, including (but not limited to) arrays, structures, unions, pointers, and functions; • Within this constraint, make use of some of the standard POSIX-compliant services;

	• Use Linux-style Makefiles to manage complex multi-file source code compilation and linking;
2. Learn how to prevent programming errors by planning application programs, and demonstrate the use of both proper design techniques and debugging tools	• Use basic analysis and design principles to describe the problem solution thoroughly, with a suitable test regimen, before beginning to code and test; • Make use of diagnostic warning and error messages, inserted output statements, and debugging tools such as gdb and ddd to correct errors of syntax and logic;
3. Acquire a solid understanding of standard C library I/O, POSIX system I/O, and the internal structure of a typical file system	• Use and understand the standard library I/O (stream) functions for stdin, stdout, and stderr, as well as other files; • Make use of POSIX-compliant system I/O functions via kernel API system calls; • Understand the structure of a typical file system (i.e., Linux ext2) including i-nodes and directories, as well as both hard and soft file links; • Redirect file I/O by manipulating the system-created file table;
4. Know when and how to create and control processes in a Unix/Linux environment	• Create and manage processes and sub-processes with POSIX-compliant system calls; • Invoke other programs in a variety of ways in newly-created sub-processes;
5. Appropriately use static, automatic, and dynamic memory	• Choosing a suitable memory allocation scheme for a program requires a good comprehension of basic memory management;
6. Use software interrupts (signals) to allow communication between processes, as well as between the console and processes	• Develop facility with the POSIX signalling functions, their required structures, and user interrupt handling functions while being able to recognize and understand older styles;
7. Use named and unnamed pipes to pass data among user-written and system-provided processes	• Pipes of both types are used to pass streams of data between pairs of processes, and require knowledge of the limitations and differences of these general types;
8. Acquire knowledge beyond the extensive online lecture notes, lab exercises, and practical assignments	• Use the required textbooks, for both required reading and self-study, online “man” pages, and the Internet to gain information.

LEARNING RESOURCES

Required Textbooks:

1) Advanced Programming In UNIX Environment, 2nd ed., by Richard Stevens and Stephen Rago, Addison-Wesley, ISBN: 0-201-43307-9

2) The C Programming Language, 2nd ed., by Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988, ISBN 0-13-110362-8

Suggested Reading:

1) The Practice of Programming, by Brian W. Kernighan and Rob Pike, Addison-Wesley, 1999, ISBN 0-201-61586-X

2) How Not to Program in C++, by Steve Oualline, No Starch Press 2003, ISBN 1-886411-95-6

3) Some useful web references:
<http://c-faq.com/> (C-faq)
<http://www.dinkumware.com/c99.aspx> (C-library}
<http://gcc.gnu.org/onlinedocs/> (GNU docs)

LEARNING ACTIVITIES

During this course, you are likely to experience the following learning activities:
- lectures that include detailed analyses of different aspects of Linux
- laboratory work
- practical and reading assignments
- research of course-related material

The course consists of 2 hours of lectures and 2 hours of lab per week, plus one hybrid hour. It is anticipated that students will need to spend an additional 5 hours per week, on average, for assignments and study.

Lectures:

Lectures will present the theoretical material of the course. Students are expected to attend all of the lectures, to take lecture notes for study and review purposes, and to read and understand applicable material in the textbooks.

Students are encouraged to ask questions during lectures and to consult with the professor on topics which they do not clearly understand. The professor will inform students, at the beginning of the course, of suitable times for consultations.

Students are expected to read and understand specific sections of the textbook. Some of this material may not be directly covered in class lectures. Students will be expected to ask for clarification and explanations as required.

Labs:

Students are expected to perform initial analysis and design before their scheduled lab in order to take advantage of the limited lab time. Laboratory assignments will be done in a lab equipped with personal computers using the Linux operating system. Laboratories will include practical programming assignments closely integrated with the lecture material.

Students are expected to work in the lab and to attempt to solve problems encountered on their own or with assistance from other students. When a difficult problem arises, the professor will try to assist the student in finding resources and developing strategies for solving the problem.

Students are expected to complete all required assignments in a satisfactory manner and demonstrate their competence in the lab exercises.

The students' ability to successfully complete the assignments and exercises will directly correlate with their level of success on tests and the final exam.

EVALUATION/EARNING CREDIT

The following will provide evidence of your learning achievements:	**This activity validates the following Course Learning Requirements and/or Essential Employability Skills:**
Theory portion: Term tests (usually 2): 35%	• Acquire a solid understanding of standard C library I/O, POSIX system I/O, and the internal structure of a typical file system - [CLR 3] • Know when and how to create and control processes in a Unix/Linux environment - [CLR 4] • Appropriately use static, automatic, and dynamic memory - [CLR 5] • Use software interrupts (signals) to allow communication between processes, as well as between the console and processes - [CLR 6] • Use named and unnamed pipes to pass data among user-written and system-provided processes - [CLR 7] • Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. - [EES 1] • Apply a systematic approach to solve problems. - [EES 4] • Use a variety of thinking skills to anticipate and solve problems. - [EES 5] • Manage the use of time and other resources to complete projects. - [EES 10]
Theory portion: Final examination: 25%	• Acquire a solid understanding of standard C library I/O, POSIX system I/O, and the internal structure of a typical file system - [CLR 3] • Know when and how to create and control processes in a Unix/Linux environment - [CLR 4] • Appropriately use static, automatic, and dynamic memory - [CLR 5] • Use software interrupts (signals) to allow communication between processes, as well as between the console and processes - [CLR 6] • Use named and unnamed pipes to pass data among user-written and system-provided processes - [CLR 7] • Acquire knowledge beyond the extensive online lecture notes, lab exercises, and practical assignments

	- [CLR 8] • Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. - [EES 1] • Apply a systematic approach to solve problems. - [EES 4] • Use a variety of thinking skills to anticipate and solve problems. - [EES 5] • Manage the use of time and other resources to complete projects. - [EES 10]
Practical portion: lab exercises (15%), assignments (usually 4 - a total of 25%): 40%	• Develop complete programs using top-down design and structured programming techniques - [CLR 1] • Learn how to prevent programming errors by planning application programs, and demonstrate the use of both proper design techniques and debugging tools - [CLR 2] • Acquire a solid understanding of standard C library I/O, POSIX system I/O, and the internal structure of a typical file system - [CLR 3] • Know when and how to create and control processes in a Unix/Linux environment - [CLR 4] • Appropriately use static, automatic, and dynamic memory - [CLR 5] • Use software interrupts (signals) to allow communication between processes, as well as between the console and processes - [CLR 6] • Use named and unnamed pipes to pass data among user-written and system-provided processes - [CLR 7] • Acquire knowledge beyond the extensive online lecture notes, lab exercises, and practical assignments - [CLR 8] • Communicate clearly, concisely and correctly in the written, spoken and visual form that fulfills the purpose and meets the needs of the audience. - [EES 1] • Apply a systematic approach to solve problems. - [EES 4] • Use a variety of thinking skills to anticipate and solve problems. - [EES 5] • Manage the use of time and other resources to complete projects. - [EES 10]

COLLEGE GRADING NUMERICAL EQUIVALENT TABLE

Final Grade	Mark Equivalent	Numeric Value	Final Grade	Mark Equivalent	Numeric Value
A+	90-100%	4.0	C+	67-69%	2.3
A	85-89%	3.8	C	63-66%	2.0
A-	80-84%	3.6	C-	60-62%	1.7
B+	77-79%	3.3	D+	57-59%	1.4
B	73-76%	3.0	D	53-56%	1.2
B-	70-72%	2.6	D-	50-52%	1.0
			F	0-49%	0
			FSP	0	0

PRIOR LEARNING ASSESSMENT AND RECOGNITION

See College Directive E35 for details on eligibility and process.

For this course, evidence of learning achievement for PLA candidates will include the successful completion of:

- A challenge exam with a breadth of coverage and level of difficulty equivalent to the final examination in the course;
- A hands-on or practical component to ensure that the requisite skill level has been achieved; and
- A computer programming (where applicable) assignment comparable to a representative assignment in the course.

RELATED INFORMATION

The following information is course-specific:

The following information is course-specific:

Assessment of student learning will be done by means of class tests, a final exam, and in-lab exercises and laboratory assignments.

Laboratory attendance is optional; however, in-lab exercises and lab assignments will form the practical portion of the final mark.

Each laboratory assignments must be successfully completed in order to obtain course credit. Late assignments will be penalized and receive a mark of zero. Any missed evaluation points will result in a grade of "0". In the case of a documented emergency the professor, in consultation with the Chair, will determine how the marks will be made up and/or final grade adjusted.

The Computer Studies Department requires that all course assignments (homework exercises, laboratory work, projects, etc) be submitted by students using a standard which could be specific to one or more courses. Professors will ensure, at the beginning of the term, that students are advised of the exact details of these course specific submission requirements. Professors will also post them online alongside the course outline. Student submissions that do not meet the course published submission standards may not be marked, and may incur a penalty of up to 100% of the submission mark.

In order to pass the course, the student must have a grade of at least 50% or "D-" on each of the theory portion and the practical portion.

All students are required to write the final exam. There are no provisions for "making up" a missed final exam. If, as a result of being off-track in your program or some unforeseen circumstance, you note that there is a scheduling conflict in your final exam schedule, it is your responsibility to alert your course professor no later than one week before final exams start, to allow for any special arrangements.

The following information is program-specific:

The following information is school/department-specific:

Retention of course material . It is your responsibility to retain copies of all assignments, labs and mid-term tests (returned from the professor), and any other evaluations and pertinent records (except for final exams, which are not returned) in case you become involved in an appeal hearing at a later date.

It is also your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.

See College Directives E15 or E24 for details in your Instaguide.

Harassment/Discrimination/Violence will not be tolerated. Any form of harassment (sexual, racial, gender or disability-related), discrimination (direct or indirect), or violence, whether involving a professor and a student or amongst students, will not be tolerated on the college premises. Action taken will start with a formal warning and proceed to the full disciplinary actions as outlined in Algonquin College Directive - A8.

Harassment means one or a series of vexatious comment(s) (whether done verbally or through electronic means), or conduct related to one or more of the prohibited grounds that is known or ought reasonably to be known to be unwelcome/unwanted, offensive, intimidating, derogatory or hostile.

This may include, but is not limited to: gestures, remarks, jokes, taunting, innuendo, display of offensive materials, offensive graffiti, threats, verbal or physical assault, stalking, slurs, shunning or exclusion related to the prohibited grounds.

For further information, a copy of the official policy statement can be obtained from the Student Association.

Violation of the Copyright Act

- General – The Copyright Act makes it an offence to reproduce or distribute, in whatever format, any part of a publication without the prior written permission of the publisher. For complete details, see the Government of Canada website at http://www.cb-cda.gc.ca/info/act-e.html . Make sure you give it due consideration, before deciding not to purchase a textbook or material required for your course.

- **Software Piracy** - The Copyright Act has been updated to include software products. Be sure to carefully read the licensing agreement of any product you purchase or download, and understand the term and conditions covering its use, installation and distribution (where applicable). Any infringement of licensing agreement makes you liable under the law.

Disruptive Behaviour is any conduct, or threatened conduct, that is disruptive to the learning process or that interferes with the well being of other members of the College community. It will not be tolerated.

Members of the College community, both students and staff, have the right to learn and work in a secure and productive environment. The College will make every effort to protect that right.

Incidents of disruptive behaviour must be reported in writing to the departmental Chair as quickly as possible. The Chair will hold a hearing to review available information and determine any sanctions that will be imposed. Disciplinary hearings can result in penalties ranging from a written warning to expulsion.

For further details, consult the Algonquin College Directive - E27 in your Instaguide.

The following information is College-wide:

Email
Algonquin College provides all full-time students with an e-mail account. This is the address that will be used when the College, your professors, or your fellow students communicate important information about your program or course events. It is your responsibility to ensure that you know how to send and receive e-mail using your Algonquin account and to check it regularly.

Centre for Students with Disabilities (CSD)
If you are a student with a disability, it is strongly recommended that you identify your needs to the professor and the Centre for Students with Disabilities (CSD) by the end of the first month of the semester in order that any necessary support services can be arranged for you.

Academic Integrity
Adherence to acceptable standards of academic honesty is an important aspect of the learning process at Algonquin College. Academic work submitted by a student is evaluated on the assumption that the work presented by the student is his or her own, unless designated otherwise. For further details consult Algonquin College Directives
E16 (http://www.algonquincollege.com/directives/sectionE/E16.pdf)
and E43 (http://www.algonquincollege.com/directives/sectionE/E43.pdf).

Course Assessments
It is Algonquin College's policy to give students the opportunity to complete a course assessment survey in each course that they take which solicits their views regarding the curriculum, the professor and the facilities. For further details consult Algonquin College Directive E38 (http://www.algonquincollege.com/directives/sectionE/E38.pdf).

Use of Electronic Devices
With the proliferation of small, personal electronic devices used for communications and data storage, Algonquin College believes there is a need to address their use during classes and examinations. During classes, the use of such devices is disruptive and disrespectful to others. During examinations, the use of such devices may facilitate cheating. For further details consult Algonquin College Directive E39 (http://www.algonquincollege.com/directives/sectionE/E39.pdf).

Transfer of Credit
Students, it is your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.