SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: COMPUTER SYSTEMS III

Code No.: CET303-6 Semester: 5

Program: COMPUTER ENGINEERING TECHNOLOGY

Author: TYCHO BLACK

Date: Sept, 1994 Previous Outline Dated: Jan., 1994

APPROVED: L. Wright 94-08-27

Dean Date

COMPUTER SYSTEMS III — CET303-6
COURSE NAME — CODE NO.

TOTAL CREDITS: 6

PREREQUISITES: CET220

I. PHILOSOPHY/GOALS:

Computer Systems III follows two years of study in a variety of programming languages and operating systems and is intended to develop the student's skills as proficient users of VAX computers at the system's programming level. Two operating systems will be used: VMS and ULTRIX. The student will learn terminal and disk I/O techniques, screen management, process management and interprocess communications. The view of the VAX computer as the host of a single, dedicated system in an environment such as process control engineering will be developed. The concepts will be demonstrated by significant programming exercises in Fortran and C.

II. STUDENT PERFORMANCE OBJECTIVES (OUTCOMES):

Upon successful completion of this course the student will:

1. Describe the services available through VMS System Services, Run Time Library and RMS.
2. Manage VMS process states and synchronize processes using event flags and other techniques.
3. Utilize mailboxes, global sections and shared files in VAX programs to demonstrate interprocess communication techniques.
4. Describe the VAX I/O system, including basic screen management and RMS and utilize their services in programs.

5. Write C shell script files.

6. Describe the nature of UNIX files and write C programs using them.

7. Manage process states in UNIX C programs.

8. Utilize UNIX interprocess communication techniques in C programs.

III. TOPICS TO BE COVERED:

1. Introduction to VMS System Services, Run Time Library and RMS.

2. Managing process states and synchronizing processes using event flags and other techniques.

3. Interprocess communication techniques in VMS using mailboxes, global memory and shared files.

4. VAX Input/Output System overview and RMS services.

5. C Shell script file basics and C programming tools.

6. UNIX regular file management.

7. UNIX process management system calls and their utilization in C programs.

8. UNIX Interprocess communication techniques including signals, pipes, named pipes and sockets.

COMPUTER SYSTEMS III — COURSE NAME
CET303-6 — CODE NO.

IV. LEARNING ACTIVITIES/REQUIRED RESOURCES

1.0 Introduction to VMS System Services, Run Time Library and RMS.

Learning Activities:

1.1 Listen to a presentation on the general capabilities of the VAX Run-time Library, System Services and RMS, and the techniques for calling them.

1.2 Review VMS concepts and architecture.

1.3 Use the VMS manuals and on-line Help facilities available as support for system-level programming activities.

Resources:

"Programming VMS in Fortran" Notes.
VAX/VMS Manuals.

2.0 Managing and Synchronizing VMS Processes.

Learning Activities:

2.1 Listen to a presentation on the types of VMS processes, the states they may be in and the methods of creating processes from DCL or programs.

2.2 Study sample VAX Fortran programs which demonstrate process management.

2.3 Write Fortran programs to determine and modify process states, and to hibernate, wake, suspend and resume processes.

2.4 Utilize the VMS Monitor and other utilities to monitor the states of processes.

2.5 Listen to a presentation on the use of event flags as process synchronization tools.

2.6 Listen to a presentation on the use of AST's, Asynchronous System Traps.

2.7 Listen to a presentation on the VMS Lock Manager.

2.8 Listen to a presentation on exception and condition handlers in the VMS environment.

2.9 Study sample programs which demonstrate process synchronization.

2.10 Complete assigned problems requiring the use of VMS facilities for managing and synchronizing processes.

Resources:

"Programming VMS in Fortran" Notes.
VAX/VMS Manuals.

3.0 Interprocess Communication Techniques in VMS.

Learning Activities:

3.1 Listen to a presentation on the use of mailboxes in VMS.

3.2 Listen to a presentation on global memory, the types of global sections and the services that can be used to call them and then write programs to utilize them.

3.3 Study sample Fortran programs which demonstrate the use of interprocess communication techniques.

3.4 Write suites of Fortran programs utilizing interprocess communication techniques.

Resources:

"Programming VMS in Fortran" Notes.
VAX/VMS Manuals.

4.0 VAX Input/Output System and RMS services.

Learning Activities:

4.1 Listen to a presentation on RMS file organization (Sequential, Relative and Indexed files) and file-sharing techniques.

4.2 Listen to a presentation on the structure of the VAX I-O system and the process of performing input and output, in particular using QIO calls.

4.3 Study sample programs which utilize QIO calls.

4.4 Write programs which incorporate low-level QIO system calls.

4.5 Listen to a presentation on the use of RTL screen management routines.

4.6 Write programs which illustrate the use of basic screen management routines to create menu systems, for example.

Resources:

"Programming VMS in Fortran" and Screen Management Notes.
VAX/VMS Manuals.

5.0 C Shell Script Files.

Learning Activities:

5.1 Review basic UNIX commands and utilities.

5.2 Listen to a presentation on C shell script files to automate common tasks.

5.3 Listen to a presentation that describes the tools that support C program development including compilation, debugging, maintaining libraries, profiling and source code control.

5.4 Listen to a presentation on the use of perror(), exit(), etc. in error handling.

Resources:

"UNIX FOR PROGRAMMERS AND USERS" by Graham Glass

6.0 UNIX File Handling

Learning Activities:

6.1 Listen to a presentation on UNIX file management system calls.

6.2 Study sample programs which demonstrate the use of UNIX I/O using file descriptors.

6.3 Write C programs which demonstrate the use of UNIX file handling.

Resources:

"UNIX FOR PROGRAMMERS AND USERS" by Graham Glass, Chap. 6

7.0 UNIX Process Management.

Learning Activities

7.1 Listen to a presentation on the UNIX environment from a process point of view and the use of system calls such as **exec, fork, exit, nice and wait.**

7.2 Write C programs that demonstrate process creation and management.

7.3 Listen to a presentation on the use of various system calls for finding and setting process identification values: **getgid, getegid, getpid, getpgrp, getppid, setuid, setgid, and setpgrp.**

7.4 Study the organization of various programs demonstrating the use of process management.

Resources:

"UNIX FOR PROGRAMMERS AND USERS", chap. 9,10

COMPUTER SYSTEMS III — CET303-6
COURSE NAME — CODE NO.

8.0 **UNIX Interprocess Communication.**

Learning Activities:

8.1 Listen to a presentation on the various types of signals used between processes in UNIX, including the **kill, pause and alarm** system calls.

8.2 Analyze and write C programs which implement signals and manage process states.

8.3 Listen to a presentation on the use of pipes and named pipes (or FIFO's) in UNIX.

8.4 Analyze and write C programs which use pipes and FIFO's.

8.5 Listen to a presentation on the various types of sockets and their use in interprocess communication.

8.6 Study C programs utilizing sockets.

8.7 Write C programs which use sockets for interprocess communication.

Resources:

"UNIX FOR PROGRAMMERS AND USERS", Chap. 10

V. EVALUATION METHODS:

3 THEORY TESTS 2 VMS Tests @ 18% each 1 UNIX Test @ 24%	60%
5 ASSIGNMENTS @ 6% each	30%
Quizzes	10%
	100%

(The percentages shown above may vary slightly where circumstances warrant.)

COMPUTER SYSTEMS III — CET303-6
COURSE NAME — CODE NO.

ASSIGNMENTS:

All assigned work must be completed satisfactorily to pass this course. A penalty of 10% per week will be applied to assignments handed in late unless extenuating circumstances exist.

It is acceptable that students consult with each other in relation to their assigned problems. However, it is unacceptable to copy programs written by someone else and submit them as your own work. Where plagiarism or copying is found and it is impossible to determine whose original work it is, a mark of zero will be assigned to all assignments involved.

GRADING SCHEME

A+	90	-	100%
A	80	-	89%
B	70	-	79%
C	55	-	69%
I	Incomplete		
R	Repeat		

UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when a student meets the following criteria:

1. The students attendance has been satisfactory.
2. An overall average of at least 40% has been achieved.
3. The student has not had a failing grade in all of the theory tests taken.
4. The student has made reasonable efforts to participate in class and complete assignments.

VI. PRIOR LEARNING ASSESSMENT:

Students who wish to apply for advanced standing in the course should consult the instructor. Prior learning assessment is not presently available for this course.

COMPUTER SYSTEMS III — CET303-6
COURSE NAME — CODE NO.

VII. REQUIRED STUDENT RESOURCES:

TEXT BOOKS:

1. "PROGRAMMING VMS IN FORTRAN" (Printed Notes)
2. "UNIX FOR PROGRAMMERS AND USERS"
   by Graham Glass (Prentice-Hall)

VIII ADDITIONAL RESOURCE MATERIALS AVAILABLE:

Book Section

DEC VMS and ULTRIX Manuals (to be itemized during the course)

IX. SPECIAL NOTES:

Students with special needs (eg. physical limitations, visual or hearing impairments, or learning disabilities) are encouraged to discuss any required accommodations confidentially with the instructor.

Your instructor reserves the right to modify the course as deemed necessary to meet the needs of students or take advantage of new or different learning opportunities.