

--	--	--	--	--	--	--	--	--	--

Sixth Semester B.E. Degree Examination, June-July 2009

Unix Systems Programming

Time: 3 hrs.

Max. Marks:100

Note: 1. Answer any FIVE full questions, selecting at least TWO questions from each part.
2. Programs must be neatly documented.

PART – A

- 1 a. Bring out the importance of standardizing the Unix operating systems. What aspects of C programming language have been standardized in ANSI C? With suitable examples, bring out the two important differences between K and R C and ANSI C with respect to function prototyping and pointers to functions. (08 Marks)
- b. What do you understand by the term feature test macros? List all the five feature test macros along with their meanings. (06 Marks)
- c. Write a C++ program to list the actual values of the following system configuration limits on a given Unix OS.
 - i) Maximum no. of child processes that can be created.
 - ii) Maximum no. of files that can be opened simultaneously.
 - iii) Maximum no. of message queues that can be accessed. (06 Marks)
- 2 a. What are APIs? When do you use them? Why are the APIs more time consuming than the library functions? (03 Marks)
- b. What are the API common characteristics? List any four values of the global variable errno along with their meaning wherever the APIs fail. (05 Marks)
- c. List all the file attributes along with their meaning. Which of these attributes can't be changed and why? List the commands needed to change the following file attributes.
 - i) File size; ii) User ID; iii) Last access and modification time; iv) Hard link count. (05 Marks)
- d. What is an inode? Why are the inodes unique only within a file system? How does OS map the inode to its filename? Bring out the four important differences between soft and hard links. (07 Marks)
- 3 a. List the structure used to query the file attributes in Unix. Write a program in C++ to list the following file attributes of a given regular file passed as command line argument.
 - i) File type ii) Hard link count iii) File size iv) File name. (08 Marks)
- b. Describe the open API, clearly indicating its prototype declaration, the values the arguments take along with their meaning. Give two instances, when open API can fail. List all the access modifier flags and explain their meanings. (06 Marks)
- c. List the important uses of fcntl API. Give its prototype description. Write a C++ program to check whether the close – on – exec flag is set for a given file. If it is not set, use fcntl to set this flag. Also show the implementation of dupz macro using this API. (06 Marks)
- 4 a. Bring out the importance of locking files. What are mandatory and advisory locks? Why is advisory lock considered safe? What is the drawback of advisory lock? Explain in brief. (05 Marks)
- b. In a certain application, it is required to lock the hatched portion of the file as shown in Fig.4(b). Before locking the program must query the OS to see if some other process has locked the file. If yes, give the details of the locked portion and the PID of the process. Once the lock is obtained perform a write and unlock the file. Write a C++ program to implement this application. Assume suitable lock type. (06 Marks)

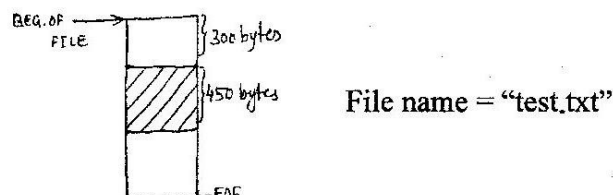


Fig.4(b). Region to be locked.

- c. What are the different ways in which a process can terminate? With a neat block schematic, explain how a process is launched and terminated clearly indicating the role of C – startup routine and the exit handlers. (05 Marks)
- d. With a neat diagram, explain the memory layout of a C program. In which segments are the automatic variables and dynamically created objects are stored? (04 Marks)

PART - B

- 5 a. With a prototype description of fork, explain the special features of this API. Write a program to create a child process and print the PPID and PID in the child process. The parent process must ensure that the child doesn't become a zombie process. The parent process must wait for the child and print exit status of the child using appropriate macros. (06 Marks)
- b. Explain in brief, what happens when exec is called in a child process. List the 6 different forms of exec APIs. Write a program that execs a program echoall to display all the command line and environment variables when this program is execed in the child process space. (06 Marks)
- c. With a neat block schematic, explain the terminal login process in BSD Unix. What is a session? Explain how do you create a session using appropriate shell commands. (04 Marks)
- d. What is job control? What are the three forms of support from the OS required for job control? (04 Marks)
- 6 a. What are signals? Mention the different sources of signals. What are the three dispositions the process has when signals occur? List any four signals along with one or two line explanation. Write a program to setup signals handlers for SIGINT and SIGACARM signals. (08 Marks)
- b. Describe the API used to mask the signals. What are signal sets? List the functions that are used to manipulate the signal sets. Write a program to demonstrate the use of sigprocmask and sigpending functions. (06 Marks)
- c. What are daemon processes? Enlist their characteristics. Also write a program to transform a normal user process into a daemon process. Explain every step in the program. (06 Marks)
- 7 a. What are pipes? What are their limitations? Write a C program that sends "Hello world" message to the child process through the pipe. The child on receiving this message should display it on the standard output. (06 Marks)
- b. With a neat block schematic, explain how FIFO can be used to implement client – server communication model. (04 Marks)
- c. What are the three different ways in which the client and server processes can get access to same IPC structure? List the APIs along with their argument details that are used to create, control, send and receive messages from a message queue. (07 Marks)
- d. What are semaphores? What is their purpose? List and explain the APIs used to create and control the semaphores. (03 Marks)
- 8 a. What is a socket? Describe the socket API. Write a C program to illustrate the process of creating socket, initializing the socket address structure and establishing a connection from a client to the server. Assume the server IP address as 10.10.2.5 and port number = 8000. The client after establishing a connection, should send "Hello World" message and wait for a reply. (08 Marks)
- b. What is out – of – band data? Illustrate with an example, how would you specify out – of – band data. (03 Marks)
- c. Write short notes on the following:
 - i) Sigsetjmp and Siglongjmp
 - ii) Race conditions;
 - iii) Error logging facility in BSD Unix. (09 Marks)

* * * * *