|  |  |  |
| --- | --- | --- |
| [http://www.pes.edu/images/PESIT_logo.gif](http://www.pes.edu/home.aspx) | **PES Institute of Technology, Bangalore**  (Autonomous Institute under VTU, Belgaum) | **10 352** |
| **CONTINUOUS INTERNAL EVALUATION (CIE) B. E. 6th SEMESTER**  **(Even Semester Jan-June 2013) TEST – 1**  **10CS 352– Unix System Programming (CS Only)** | | |
| Time: 1½ Hrs Answer All Questions Max Marks: 50 | | |

Genreal Instructions:

* Answer the questions precisely.
* **Write program snippets. You do not have to write the whole program**. Assume that the required headers are available,

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | a  b | Create a file programmatically with the permission rw-rw-rw-. Take care of umask programmatically.  **umask(0);**  **int fd = creat("myfile.txt", S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IWGRP | S\_IROTH | S\_IWOTH);**  **if(fd < 0)**  **{**  **perror("creat"); exit(1);**  **}**  **close(fd);**  What is the difference between sticky directory and sticky file?  **In a sticky directory, only the owner of the file can remove the file.**  **A sticky file should be an image and once loaded gets stuck into memory and there is no loading of the file each time the same image is executed** | 6  4 |
| 2. | a  b | How can the user change his password by using the program /usr/bin/passwd when the passwords are stored in a file (/etc/shadow) which cannot be accessed by the user?  **/usr/bin/passwd has set user bit – is owned by the root – the super user. Whenever this program is run by the user, he becomes as powerful as the owner of the file. Super user being the most trusted user can modify any file. Thus, a user can change his password.**  Given fd, file descriptor to a file opened for reading, write a program to read the last n characters of the file and display it.  ***long n;***  ***scanf("%ld", &n);***  ***char str[n + 1];***  ***int size;***  ***if(lseek(fd, -n, SEEK\_END) < 0)***  ***{***  ***perror("lseek"); exit(1);***  ***}***  ***if((size = read(fd, str, n)) < 0)***  ***{***  ***perror("read"); exit(2);***  ***}***  ***str[size] = '\0';***  ***printf("str : %s\n", str);*** | 4  6 |
| 3. | a  b  c | Write a program to delete a file if it is empty.  ***struct stat s;***  ***if(stat("filename", &s) < 0)***  ***{***  ***perror("stat"); exit(1);***  ***}***  ***if(s.st\_size == 0)***  ***if(unlink("filename") < 0)***  ***{***  *perror("unlink");*  *}*  Why all permissions are enabled on a symbolically linked file?  ***Symbolically linked file forwards most of the commands to the file which it refers to. It is not supposed to block these commands.***  The file myprog is an image obtained by compilation and linking. While executing the program  ./myprog will work and  Myprog may not work. Why?  ***PATH variable is used to locate the file if there is no / in the command. In the first case, shell looks for the file in the current directory. In the second case, the command would work if current directory is in the PATH.*** | 6  2  2 |
| 4. | a  b | What happens to the following on a call to one of the exec family of functions with respect to the process calling exec?  Pid : ***same as the execing process***  Ppid ***same as the execing process***  umask value ***same as the execing process***  address of global variable :***not meaningful as a new process overlays the data segment***  open file descriptor : **remains open unless close\_on\_exec flag is set by the execing process**  What does the system call wait return? When does wait return immediately? What is the argument passed to wait? How is this populated on return?  ***Wait returns the pid of the child whose status is being checked***  ***Wait returns immediately if any of children are in a zombie state – has completed execution***  ***Pointer to an int variable***  ***Least significant byte: 0 on normal exit of the child or signal number if killed by a signal number***  ***Next to least significant byte: exit value on normal exit*** | 5  1 + 1 + 1 + 2 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| 5. | a  B  c | What is the difference between an orphan and a zombie?  What resources are held back by a zombie?  ***An orphan process has lost its parent and now reports to the process init whose pid is 1***  ***A zombie is a dead process which has informed the parent and the parent is yet to check the status of the child process.***  ***A zombie occupied a slot in the process table – thus reducing the number of processes which can be launched at the same time.***   1. If a new file created gets the permissions, rw- - - - - - - , what is the value of umask? 2. What is the type of file created when a symbolically linked file is copied? 3. When does mv command require read permission on the source file? 4. ***066*** 5. ***Regular file*** 6. ***Move across file system***     Write a program snippet to check whether the file whose name is given has multiple links.  ***struct stat s;***  ***if(stat("filename", &s) < 0)***  ***{***  ***perror("stat"); exit(1);***  ***}***  ***if(s.st\_nlink > 1)***  ***printf("has more than one name");***  ***else***  ***printf("has only one name");*** | 2 + 1  3  4 |