

06/02/24

8

EXPERIMENT - 3

STUDY OF LINUX SYSTEM CALLS

PROGRAM 1

AIM

Implement a program to create a child process using system call `fork()` and to demonstrate system call `getpid()`

ALGORITHM

STEP 0: START the process

STEP 1: Declare a variable 'p' to be shared by both child and parent

STEP 2: Create a child process using `fork` system call and get its return value to 'pid'

STEP 3: If return value is -1, then

i) Print process creation failed

ii) Terminate using `exit` system call

STEP 4: If return value is 0 then

i) Print child process created

ii) Print process id using `getpid` system call

iii) call `execvp()` to execute another program ".|child"

iv) create another program child which prints

STEP 5: Otherwise,

i) Store the returned child process

ID in childpid

- ii) wait for child process to finish using wait() system call, store status to status
- iii) Print "Parent process created with ID: "
- iv) Print process ID using getpid system call.
- v) Print "child process created successfully"

STEP 6: STOP.

RESULT

child process created and parent ID displayed here. Output obtained.

PROGRAM 2

AIM

To implement a program to demonstrate system call stat.

ALGORITHM

STEP 0: START

STEP 1: Declare a character array 'file' to store filename and input from user

STEP 2: Declare a pointer to a struct stat named 'node' and allocate memory for it

STEP 3: Use stat function to retrieve

information by pairing file and store it in node.

STEP 4: Check if $\text{node} \rightarrow \text{st_ino} = 0$

i) Print 'File does not exist'

ii) go to STEP 6

STEP 5: Otherwise

i) Print Inode $\text{node} \rightarrow \text{st_ino}$

ii) Print Block size $\text{node} \rightarrow \text{st_blksize}$

iii) Print Access time $\text{node} \rightarrow \text{st_atime}$

iv) Print last modified time
 $\text{node} \rightarrow \text{st_mtime}$

v) Print group ID $\text{node} \rightarrow \text{st_gid}$

vi) Print no. of file $\text{node} \rightarrow \text{st_nre}$

vii) Print permissions $\text{node} \rightarrow \text{st_mode}$

viii) Print user ID $\text{node} \rightarrow \text{st_uid}$

STEP 6: END

RESULT

Executed program is demonstrate
that obtained output

PROGRAM 3

AIM

To implement a program to demonstrate
system calls `opendir` & `readdir`

ALGORITHM

STEP 0: START

STEP 1: Declare character array `dir-name`

STEP 2: Input directory name to
dir-name

STEP 3: Open the directory using
system calls opendir() and
store its pointer in dir

STEP 4: Repeat the steps while
ptr2 != NULL by storing value by
calling system call readdir
(dir)

Print ptr2->d_ino and
ptr2->d_name

STEP 5: Close the directory

STEP 6: END.

RESULT

Executed the program to demonstrate
system calls opendir & readdir and
obtained output.