//shell script to call following system calls

//1 exec command in Linux is used to execute a command from the bash itself. This command does not create a new process
// it just replaces the bash with the command to be executed. If the exec command is successful, it does not return to the calling process.
exec [-cl] [-a name] [command [arguments]] [redirection ...]
//2 The fork system call is used to create a new processes. The newly created process is the child process
pid_t fork(void);
//3 exit command in linux is used to exit the shell where it is currently running. It takes one more parameter as [N] a nd exits the shell with
// a return of status N. If n is not provided, then it simply returns the status of last command that is executed.
exit [n]
//4 getuid() returns the real user ID of the calling process.
uid_t getuid(void);
//5 geteuid() returns the effective user ID of the calling process.
uid_t geteuid(void);
// 6 getppid() : returns the process ID of the parent of the calling process. If the calling process was created by the for k() function and the
// parent process still exists at the time of the getppid function call, this function returns the process ID of the parent process
pid_t getppid(void);
//7 getpid() : returns the process ID of the calling process. This is often used by routines that generate unique tempor ary filenames
pid_t getpid(void);
//8 kill -l :To display all the available signals you can use below command option:
$kill -l
//9 signal() sets the disposition of the signal signum to handler.
// Use the chdir command to change to another directory.
chdir /home/rich/www