UNIX SYSTEM PROGRAMMING

File I/O

Chandravva Hebbi
Department of Computer Science and Engineering
chandravvahebbi@pes.edu

UNIX SYSTEM PROGRAMMING

File I/O

Chandravva Hebbi
Department of Computer Science and Engineering

UNIX SYSTEM PROGRAMMING

Topics to be Covered

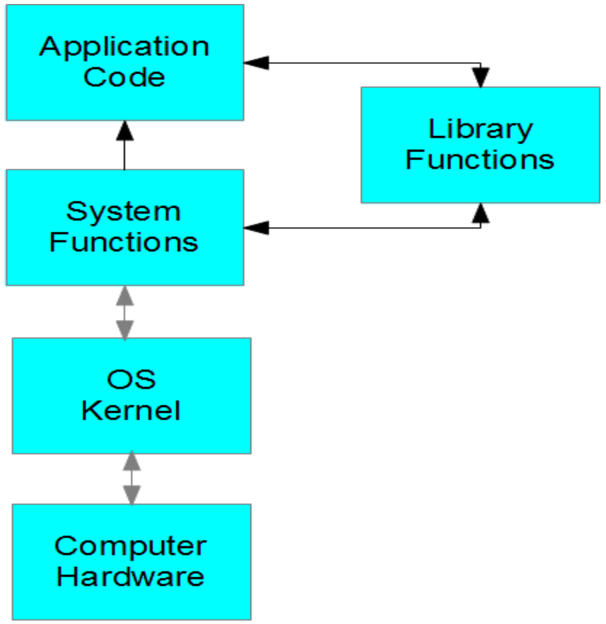
❖System calls

❖Kernel Data structures

UNIX SYSTEM PROGRAMMING

Library and System calls

- The functions which are a part of standard C library are known as **Library functions**.
- The functions which change the execution mode of the program from user mode to kernel mode are known as **system calls.**
- The system calls are required when the user programs need services from the OS.



Why System Calls

- System calls acts as entry point to OS kernel.
- There are certain tasks that can only be done if a process is running in kernel mode.
- Examples of these tasks can be interacting with hardware etc.

Types of Library Functions.

- Functions which do not call any system call.
- Functions that make a system call

Sysem Calls

There are 5 basic system calls that Unix provides for file I/O.

1. int open(char *path, int flags [, int mode]);
2. int close(int fd);
3. int read(int fd, char *buf, int size);
4. int write(int fd, char *buf, int size);
5. off_t lseek(int fd, off_t offset, int whence);

Open Calls

int open(char *path, int flags [, int mode]);

Open makes a request to the operating system to use a file.

- ❖ 'path' argument specifies what file you would like to use
- ❖ 'flags' and 'mode' arguments specify how you would like to use the file.
- ❖Return: File desciptor, -1.

Open Call

open(filename, flags, mode)

Returns lowest numbered available file descriptor

Filename

Flags for mode of access

O_RDONLY

O_WRONLY

O_RDWR

Other flags

O_CREAT - Mode to be specified if file is to be created

O_APPEND

O_TRUNC

O_EXCL

O_NONBLOCK,O_DSYNC,O_RSYNC,O_SYNC

The creat() system call

creat(pathname,mode)

Returns the file descriptor opened for write-only

Returns -1 on error

Equivalent to :-

open(pathname,O_WRONLY|O_CREAT|O_TRUNC,mode)

File is opened only for writing

Before new version of open, to write and read,

had to call creat, then open.

Better way :-

open(pathname,O_RDWR|O_CREAT|O_TRUNC,mode)

The close() system call

close(filedes)

Returns 0 if OK, -1 on error

Closing releases any record locks that process may have on the file

When process terminates, all of its open files are closed automatically by the kernel.

The read() and Write system calls

Different from fread() and fwrite()
Uses fd returned from open() call
Take three arguments, fd, Buffer to be read into or written from
Count.

lseek() system call

Used to position the file pointer randomly
Next I/O proceeds from that point onwards
lseek(fd, offset, origin)
Origin
0 - beginning of file
1 – current position
2 – end of file
SEEK_SET, SEEK_CUR, SEEK_END

THANK YOU

Chandravva Hebbi

Department of Computer Science and Engineering