Tinf

SYSTEM CALLS:

fork(): creates a new process by duplicating the calling process, the new process is referred to as the child process,

the calling process is referred to as the parent process

stat: stat() is a Unix system call that returns file attributes about an inode. The semantics of stat() vary between operating systems. As an example, Unix command ls uses this system call to retrieve information on files that includes: atime: time of last access (ls -lu),mtime: time of last modification (ls -l),ctime: time of last status change (ls -lc)

lstat(): lstat() is identical to stat(), except that if path is a symbolic link, then the link itself is stat-ed, not the file that it refers to

fstat(): fstat() is identical to stat(), except that the file to be stat-ed is specified by the file descriptor fd

kill():In computing, kill is a command that is used in several popular operating systems to send signals to running processes in order to request the termination of the process,

the default signal for kill is TERM

sigkill: send signal

mmap(): creates a new mapping in the virtual address space of the calling process,the starting address for the new mapping is specified in addr,

the length argument specifies the length of the mapping.

munmap():the munmap() system call deletes the mappings for the specified address range, and causes further references to addresses within the range to generate invalid memory references.

chmod(): changes the mode of the file specified whose pathname is given in pathname, which is dereferenced if it is a symbolic link.

fchmod(): changes the mode of the file referred to by the open file descriptor fd.

waitpid(): all of these system calls are used to wait for state changes in a child of the calling process, and obtain information about the child whose state has changed, a state change is considered to be: the child terminated; the child was stopped by a signal; or the child was resumed by a signal,

suspends execution of the current process until a child specified by pid argument has changed state. By default, waitpid() waits only for terminated children, but this behaviour is modifiable via the options argument

wait(): the wait() system call suspends execution of the current process until one of its children terminates

waitid() system call (available since Linux 2.6.9) provides more precise control over which child state changes to wait for

ERROR:

fork: ENOMEN: fork() failed to allocate the necessary kernel structures because memory is tight

exec: E2BIG: the total number of bytes in the environment (envp) and argument list (argv) is too large

unlink: EBUSY: the file pathname cannot be unlinked because it is being used by the system or another process; for example, it is a mount point or the NFS client software created it to represent an active but otherwise nameless inode ("NFS silly renamed")

read: EAGAIN: the file descriptor fd refers to a file other than a socket and has been marked nonblocking (O\_NONBLOCK), and the read would block

mount: EACCES: A component of a path was not searchable

chmod: EFAULT: pathname points outside your accessible address space

kill: EFAULT: pathname points outside your accessible address space

Traps:

In computing and operating systems, a trap, also known as an exception or a fault, is typically a type of synchronous interrupt typically caused by an exceptional condition (e.g., breakpoint, division by zero, invalid memory access). A trap usually results in a switch to kernel mode, wherein the operating system performs some action before returning control to the originating process. A trap in a system process is more serious than a trap in a user process, and in some systems is fatal. In some usages, the term trap refers specifically to an interrupt intended to initiate a context switch to a monitor program or debugger