Linux Kernel and Device Drivers interview Questions and Answers - Process Management

Process Management

1) How to manipulate the current process states?

Linux kernel provide set_task_state(task,state) by which you can manipulate the state of given task. By using of set_task_state you can change to state of given task.

2) What are kernel thread?

Kernel perform some background operation by using of some thread that is called kernel thread.kernel thread are the special thread which doesnt have the process address space .They just run in the kernel space . We can reschedule kernel thread like a normal process .Kernel delegates several task to kernel thread like flush etc. Read more about kernel thread

3) How threads are implemented in linux kernel?

In Linux there is no special provision for thread . In Linux thread is a process which actual share some resources with other process . Every thread has a unique task_struct.**creating thread**Thread also created by using the clone sys call except some flags for sharing the data like file system , signal handler,open files also set. Flags are used to specify that which resources are shared between parent and child process or thread.E.g**clone(CLONE_VM | CLONE_FS | CLONE_FILES | CLONE_SIGHAND, 0);**

4) What are different states of a process in linux?

In linux a process can be in one of the following state.

1)TASK_RUNNING :- This state specify that process is either running or in ready state and waiting in runqueue .

2)TASK_INTRUPTTIBLE:-This condition states that the task is in sleeping state and waiting for some event to occur. When this condition occurs task comes in TASK_RUNNING state.

3) TASK_UNINTRRUPTIBLE:-This state is similar as previous state but in this state process does not wait for any event to occur.

4)_TASK_TRACED:-The process is getting traced by another process , such as a debugger.

5) What is difference between process and thread?

Linux does not differentiate between process and thread. Linux kernel understand a thread as a process and implements process and thread in same way.In Linux thread is a process which actually share some resources with other process .

6) Generally what resources are shared between threads?

Threads can share lots of things . Some of the most shared resources are file system information , open files , signal handlers , Address space etc.

7) What is process descriptor?

In Linux kernel a process is represented by Process descriptor which is a structure of type task_struct . Kernel stores all the process by maintaining a doubly linked list in which each node is type of task_struct. In other words we can say kernel maintained a doubly linked list of process descriptors.

8) What is task_struct?

The task_struct is a large data structure around 1.7 Kb on a 32 bit machine. This structure maintain all the information about a process. This structure contains info like signals, files ,stack ,state and much more as we can see below picture. Read more about task_struct

10) What was the need of thread_info structure?

The task_struct is a large data structure around 1.7 Kb on a 32 bit machine and kernel stack is either 4KB or 8KB . Hence the task of storing structure of 1.7 kb

is very much difficult . So kernel introduced concept of thread_info ,which is very much slimmer than task_thread and just points to task_struct structure.

11) Difference between fork() and vfork() ?

The vfork() system calls has the same effect as fork() except that the page table entries of the parent process are not copied . Instead the child executes as a sole in the parents address space, and the parent is blocked until the child either call exec() or exits.

12) What is process context ?

Normally process runs in user space but when a program calls a syscall or triggers an exception then it enter into the kernel-space at that time kernel is said to be executed on behalf of process and in other word kernel is in process context.

13) What is zombie process?

Process which is dead and completed its execution but still it had an entry in process table is called zombie process.

14) How parent less process is handles in linux ?

If a parents exist before its children then child process are re-parented to another process in same thread group or , if that fails , then **init process.**

Virtual File System

1) What is virtual file system and what is the need of it in linux.

Virtual file system is the sub system of the kernel that implements the file and provides file system related interface to user space applications. VFS provide the abstraction interface which provide a generic interface to access different types of devices based on different file system likes ext2 , ext3 etc.

2) What are different object types in VFS.

VFS has below four types of objects in Linux

- superblock
- inode
- dentry object
- file

3) What are the operations possible on inode and superblock objects.

inode specifies the operation that can be performed on a file like create() and link() etc.

superblock specifies the operations which can be performed on specific file system like create_fs() and sync_fs() etc.

Device Driver Questions

1) What is device driver and what is the need of it?

Device Driver is a program which interacts and control your hardware devices present in a system. For newly developed devices or enhancing of some feature for existing device , we need to write device driver.

2) What are different kind of devices?

There are three kind of devices available in market

1. character devices
2. block devices
3. network devices

3) What is module in linux.

Kernel module is the piece of code which can be loaded and unloaded into kernel upon demand. it means you can load your code or module into kernel whenever it required and you can unload your code and module from kernel whenever you don’s required.

4) How mudules are loaded in linux.

modules can be loaded dynamically by following command

- insmod
- modprob

5) Difference between insmod and modprobe.

Both commands are used to insert kernel module dynamically . modprobe first loads the modules on which new module has dependency but insmod doesn’t check any dependency .

6) How parameters are shared between driver modules.

different parameters can be shared by exporting parameters in global space which can be done by

EXPORT_SYMBOL(sym)

7) What are IOCTLS.

linux provide us a generic call which can be used to control any device which is called **IOCTL** or input output control. As the name suggest by using of ioctl we can write to the device as well as we can read from the device also.

8) What is syscalls

System Calls provide an interface between hardware and user space. System calls are required when we have to use any service of kernel. Suppose we want to change the date and time of system in that case we need to request to kernel through Syscalls.

9) What are the benefits of syscalls.

It serves mainly three purpose :-
1) Provide abstract hardware interface.
2) Security and stability of our system

3)It provide a ritualized system.

10) How character driver is registered in linux.

We can register driver by following function

register_chrdev(major_number,device,file_operation)

11) What is init and exit function of a driver

our driver needs to interact with kernel. So driver has to notify to kernel that it also exist in the system.So to notify to the kernel driver has init calls which is used for hooking our driver routines into kernel. Besides notifying or registering with kernel it also allocates any kernel memory required for device and initialize hardware. This function is called at boot time.

In the same way we need to detach our driver from the system when our system is shutting down . To serve this purpose driver use Exit call which is called by kernel at the time of shutting down of system. This call unregister driver or free all the resources it allocated at the time of init call.

12) How and when init and exit function of driver get called.

init function is called at the time of module load and exit is called at the time of unloading of module.

13) What is probe function?

probe function is used for platform devices . Probe function pointer is initialized in the init function . Probe function is used to initialize hardware, allocating resources and registering the devices with the kernel.

14) When probe is get called.

Whenever device name match with driver name initialized at boot time , probe function is called.

15) What is platform devices.

Platform are the devices which are non discoverable at the hardware level and connected to a virtual bus i.e platform bus.

16) What is device tree.

linux use a data structure to represent connected devices , this data structure is called device tree.

17) What are the benefits of device tree over board files.

following are the advantages of device tree

- Simple to change the configuration without compiling any source code.
- can easily add support for new hardware
- provide easy to use and easy to read description of hardware.

18) What is sysfs and procfs?

procf is a special file system which is used to avail information about processes , and to change kernel parameter at run time. it is mounted as /proc at the boot time. Its an old interface between kernel space and user space.

Sysfs is a virtual file system provided by linux kenel to expose information about Hardware devices, kernel subsystems and driver at run time. It also provide capability to pass some controlling commands to some hardware devices.

19) How logs are printed in linux kernel and what are the logs level available in linux.

In Linux kernel logs are printed by special function printk() , which is just like our printf function . There are different types of log level available in linux kernel , below is the list of Log levels

0 (KERN_EMERG) system is unusable

1 (KERN_ALERT) action must be taken immediately

2 (KERN_CRIT) critical conditions

3 (KERN_ERR) error conditions

4 (KERN_WARNING)warning conditions

5 (KERN_NOTICE) normal but significant condition

6 (KERN_INFO) informational

7 (KERN_DEBUG) debug-level messages

20) What is copy_to_user and copy_from_user.

copy_to_user is a function which is used to copy some data from kernel space to user space.

copy_from_user is a function which is used to copy data from user space to kernel space.

21) What do you mean by kernel configuration and what are the various way of configuring kernel.

Kernel which we get from Kernel.org is generic kernel which we can build and install on different hardware platform . But before building Linux kernel we need to customize or configure the Linux kernel or we can say we need to do some platform specific changes to port our linux kernel on our hardware platform.

For configuring the kernel go into root directory of Kernel uncompressed folder and use one of the following way of Kernel Configuration.

Linux kernel provides us various way of configuring the Linux Kernel. Following are some method to configure the Linux kernel

1. make config
2. make menuconfig
3. make defconfig

22) What is menuconfig

Linux provides a graphical view of configuring kernel , menuconfig. when we run “make menuconfig” command we get a GUI on which we can see multiple kernel settings.

23) What is ioremap

ioremap is the function provided by linux kernel to map physical memory into kernel virtual address space . To access any hardware device first we need to map that device into kernel virtual address then only we can access that device.

24) What is seagmentation fault.

segmentation fault is an error generated by Linux kernel , whenever any process tries to access memory location for which it doesn’t have permission to do so.

25) What are the various ways of debugging linux kernel.

There are various way of debugging linux kernel . GDB, KGDB, printk statements etc are the method of software debugging while JTAG debugging is one of the hardware debugging.

27) What is zimage and bzimage.

zImage :- Its a compressed image format of linux kernel which is self extracting.We can build it by giving following command while building kernel

make zImage

bzImage:- Its also a compressed format of linux kernel image but this image is much compressed as compare to zImage.

28) What are different booting arguments in linux.

http://man7.org/linux/man-pages/man7/bootparam.7.html

29) How parameter are passed from boot loader to kernel?

Parameters are passed to linux kernel from bootloader by ATAG.

30) What is ATAGS

ATAGS is way of sending different command at boot time to send different parameters from boot loader to Linux kernel.

31) From which file kernel execution starts.

Linux kernel execution starts from head.s file.

32) What is bootloader

Bootloader is a program or code which is responsible for early stage initialization during booting and loading linux kernel into memory.

33) What is primary and secondary bootloader

Primary bootloader is small piece of code which executes from ROM and does core initialization and pass control to secondary stage boot loader which is a program which executes from RAM and does further initialization and load linux kernel to memory.

34) Why we need two bootloader

Because primary bootloader executes from ROM and the size of ROM is too small to accommodate all the code required for initialization , hence we need secondary boot loader to perform rest of the initialization.

35) Difference between poll and select

In select you have limit on fds on which you can have a watch but in case of poll no limit on number on fds.

36) What is priority inheritance and priority inversion.

Priority inversion is a situation in which a resource is acquired by low priority process, which high priority task wants to acquire and in between some medium priority task preempts low priority task which was holding the resources.

To solve this priority inversion problem , the solution is to increase the priority of process which is holding resource to the max priority. which is called priority inheritance.

37) What are different type of kernel?

There are three types of linux kernel

Microkernel

Monolithic Kernel

Hybrid Kernel

38) What is DMA

DMA stands for Direct memory access . Its a method which allows different input output device to send and receive data directly to or from main memory by bypassing CPU to speed up the data transaction .

39) What is cache coherency

cache coherency is a way to maintain consistency of data stored or shared in different local cache.It ensures that changes in the values of shared data are propagated throughout the system in a timely fashion.

40) What is copy on write

Usually when we call the fork a new address space is created and parents data is copied into that but it’s a wastage of huge memory . To avoid this Linux implement fork() in such a way that same process data is shared between process and child until child write something new or updates . If child tries to write something then the copy of parent process data is created.

41) What is highmem and lowmem

Kernel virtual memory is divided into two part Highmem and lowmem.

Highmem is the memory which contains memory which is not permanently mapped to physical memory. It generally contains Process related stuff or kernel module specific stuff.

Lowmem is the memory section which is permanently mapped and never swaps out. It mostly contains kernel image with other core things.

42) What happens if we pass invalid address from userspace by using ioctls

If you pass invalid address from userspace to kernel then it might lead to severe problems like crash , data overriding and loosing of some data etc.

43) What are different ipc mechanism in linux

sockets , Pipes , Shared memory ,signal, message queue, semaphore , mail box

44) what are sockets?

Socket is in IPC mechanism supported by linux kernel to provide communication between different process. Its basically a logical endpoint for communication between different process.

45) How page fault is handled in linux?

Whenever the kernel tries to access an address that is currently not

accessible or that particular page is not available in memory, the CPU generates a page fault exception and calls the

page fault handler

void do_page_fault(struct pt_regs *regs, unsigned long error_code)

Page fault handler then first load the required page from some secondary memory to main memory and modifies the page tables accordingly. After modifying page tables it notifies to CPU to continue .

47) What is I2c and SPI.

The Inter-integrated Circuit (I2C) Protocol is a protocol intended to allow multiple “slave” digital integrated circuits (“chips”) to communicate with one or more “master” chips.

read more from

https://learn.sparkfun.com/tutorials/i2c

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between CPU and peripherals such as registers, sensors etc. It uses clock and data lines, along with a select line to choose the device you wish to communicate.

48) How physical to virtual translations works in linux.

phys_to_virt(phys_addr) api can be used for this purpose.

49) What is thrashing, segmentation and fragmentation.

Thrashing is a state when process of swap in and swap out of page to or from main memory is consuming most of the CPU time. This state occurs whenever successive page fualt occurs in very small duration.

As a result of dynamic memory allocation , memory is divided into chunks of free memory and there might be scenario that we are not able to satisfy memory allocation request because size of free memory chunks is too small but overall total free space is large than memory required is called fragmentation.

seagmentation is a process of dividing memory into different sections or segments or we can say a logic unit such as main program , procedure , function , method etc.

50) What is preempt_count and what is the need of that.

Preempt_count is a variable associated with each process , which maintain count of number of lock hold by particular process . Default value of preempt_count is zero. It increase by one when process acquire a lock or decrease by one when process release the lock. Linux kernel check its value before it preempt a process to make ensure process is not holding any lock.

Kernel Synchronization

1) What is synchronization

Process of limiting simultaneous access to shared resources by more than one process is called kernel synchronization. synchronization ensure that only one process will get access to shared data at a time.

2) What is critical section

it’s a part of code which cannot be accessed by more than one process at a time to avoid any inconsistency of shared data or we can say this part of code access and manipulate shared data.

3) What is race condition

When more than one process access the same critical at same time , its called race condition.

4) Why we need to take care of synchronization

If we avoid the synchronization then there are chances that shared data might be manipulated or modified by more than one process . Which can leads to inconsistency of data. read more at http://tutorialsdaddy.com/tutorials/synchronization-using-semaphore.php?course=Linux%20device%20driver

5)What is various synchronization techniques in linux.

Linux kernel provide various synchronization techniques

1. Semaphore
2. Mutex
3. Spinlock
4. read write spin lock
5. read write semaphore

6) What is deadlocks.

A deadlock is a condition involving one or more threads of execution and one or more resources, such that each thread waits for one of the resources, but all the resources are already held. The threads all wait for each other, but they never make any progress toward releasing the resources that they already hold. Therefore, none of the threads can continue, which results in a deadlock.

7) What is atomic operations.

Atomic operations provide which executes atomically , without interruption. It means atomic operations cannot be interrupted in between by some other process.

8) What is spin locks

Spin lock is one of the synchronization method in which a process has to acquire a lock before accessing critical section . If lock is already held by some

other process then our process which is requesting to enter in critical section ,has to wait for releasing of lock. Waiting process will keep on executing on the processor and will be waiting for lock release.

9) What is reader-writer spin lock.

Reader writer spin lock is an advancement to spinlock. In reader writer spin lock more than process can acquire lock at the same time if they just want to read the shared data. But Only one process will get lock if they wants to write or modify the data.

10) What is semaphore.

Semaphore is one of the synchronization technique supported by linux kernel. In semaphore only one process can get lock to access the same critical section at a time. If a process tries to acquire a lock for a critical section which is already held by some other process then our process which is requesting has to wait and it will go for sleep till the lock is released by some other process.

11) What is binary semaphore

Binary semaphores can take only 2 values (0/1). They are used to acquire locks. When a resource is available, the process in charge set the semaphore to 1 else 0.

12) What is difference between semaphore and spin lock

In semaphore , if lock is already held by some other process then process goes into sleep state till the time lock is available. But in case of spinlock process who made the request of lock will be keep on executing on processor in busy loop and will be waiting for lock to be released.

13) When to choose what among spin lock and semaphore.

If you are sure that your process will get lock in very short period of time then spinlock is recommended to avoid context switching overhead. But if your process has to wait for long to get a lock then semaphore is the preferred way of implementing synchronization.

14) What is difference between semaphore and mutex.

Both are the techniques for maintaining synchronization . Mutex ensure mutual exclusion . It means a process who acquired the lock that only can release the lock. No other high priority task can release the lock acquired by some other process in mutex. But in case of semaphore any high priority process can release the lock acquired by some low priority process.

15) What is preemption disabling and what is the use of this.

As Now linux kernel is preemptive . Any task can be preempted by some high priority task. Preemption disabling is used to disable the preemption of low priority task by other high priority task. Preemption disabling is used for implementing spinlock on uniprocessor system.

Memory Management

1) How memory is managed in linux.

Memory in linux , managed by memory management subsystem in linux kernel which takes care of all the memory related stuff like memory allocation, page allocation, virtual memory management , memory mapping etc.

2) What are pages

basic unit of memory management in linux kernel is pages. Physical memory is divided into pages which is of size 4kb in case of 32 bit system .

3) What are different memory zones in linux.

Linux kernel divides memory into four zones

ZONE_DMA: This zone contain pages that is used for DMA operations

ZONE_DMA32: This zone is just like ZONE_DMA but this can be accessed only by 32-bit devices.

ZONE_NORMAL: This Zones contain normal regularly mapped page.

ZONE_HIGHMEM: This zone contains which contains pages that are not permanently mapped into kernel’s address space .

4) How to allocated pages.

below are the APIs provided by linux kernel for allocating pages

- alloc_pages()
- get_free_page()
- get_zeroed_page()
- get_free_pages()

5) How to freeing page

Page can be freed by below APIs in linux kernel

- __free_pages()
- free_pages()
- free_page()

6) What is kmalloc and what are action modifier we can pass while using kmalloc?

kmalloc is an api in linux kernel which is used to allocate memory in Linux kernel.Action modifiers is one of the argument need to passed while allocating memory by using of kmalloc. Action modifiers specifies how the linux kernel is supposed to allocate the requested memory. below are some of the action modifiers

- __GFP_WAIT
- __GFP_HIGH
- __GFP_IO
- __GFP_KERNEL
- __GFP_DMA

7) What is zone modifier in linux.

Zone Modifier specifies from where to allocate memory. Is tells from which zone memory is to be allocated.

8) What is vmalloc

vmalloc is also an API just like kmalloc which is used to allocate memory. Vmalloc ensure memory allocated will be virtually contiguous but physical memory can ne non contiguous .

Interrupt

1) What is interrupt ?

Interrupts enable hardware to signal to processor. When CPU is busy in doing or processing something at that time any hardware can request to CPU for some service. CPU saves it current state and serves the hardware request. After full filling the request of hardware CPU restore its previous state and start processing previous that it was doing before receiving from hardware. Read more about interrupt

http://www.tutorialsdaddy.com/2015/09/interrupts/

2) What is interrupt handler or ISR ?

This is the function which kernel runs in response to a specific interrupt is called interrupt handler or ISR. This function actually handle and process the request of interrupts. Interrupt handler are mainly divided into two parts

3) What is top halves and bottom halves ?

Interrupt handling is divided into two parts1) Top Halves:- Its executed as immediate response to interrupt.2)Bottom Halves:- Its executed some time later when CPU get free time.Read more about top halves and bottom halves

http://www.tutorialsdaddy.com/2015/09/top_halves_and_bottom_halves_in_interrupt_handling/

4) How interrupt is registered ?

It is the responsibility of device driver to write the interrupt handler and take care of interrupt request generated from its device. Kernel provide below function to register an interrupt.int request_irq()

5) What are different interrupt handler flags ?

Below are the different Interrupt handler flags and their usageIRQF_DISABLED:- when this flag is set. It indicates the kernel to disable all interrupts when executing this interrupt handler.when unset , interrupt handler run with all interrupts except their own enabled.IRQF_TIMER:- this flag specify that this handler processes interrupts for the system timer.IRQF_SHARED:- This flag specifies that interrupt line can be shared among multiple interrupt handler.

6) How interrupt are freed ?

Below is the function availed by kernel to free the interrupt handler.void free_irq(unsigned int irq, void *dev)

7) What are the considerations needs to taken care while writing interrupt handler ?

In writing interrupt handler we need to think about what are the services our device may request. kernel provide below syntax of interrupt handler. While

writing interrupt handler we should not use any system calls which may lead to call sleep. As we are not allowed to call sleep while handling interrupt.

9) What is interrupt context ?

Linux Kernel execution on behalf of any interrupt is called interrupt context.

10) How to disable and enable interrupts ?

Below are the functions provided by Linux kernel for disabling and enabling of interrupts

11) What are different bottom halves techniques in linux ?

1) Original bottom Half

2)Task Queues

3) Softirq

4)Tasklets

5)workqueues

12) What is tasklets , softirq and workqueus and difference among them ?

Read

http://www.tutorialsdaddy.com/2015/10/softirq-and-tasklets/

13) When to choose which bottom halves ?

There are many considerations to be taken while deciding which bottom halves to be usedIf you don’t need to maintain any synchronization between execution of code on different processor then go for softirq but if you need to maintain sync , go for tasklets. If your code required to go into sleep state then workqueue is the available option.

14) How to implements softirq , tasklets and workqueus ?

Read

softirq and taskets work queues

http://www.tutorialsdaddy.com/2015/10/linux-interrupt-handling-work-queues/

15) How to schedule tasklet ?

Tasklets are scheduled via tasklet_schedule() which takes pointer to tasklet_struct.after tasklet is scheduled , it runs once at some time in the near future.

16) What is ksoftirqd?

read

http://www.ms.sapientia.ro/~lszabo/unix_linux_hejprogramozas/man_en/htmlman9/ksoftirqd.9.html

17) How to disable bottom halves ?

Local_bh_disable() and local_bh_enable() are the APIs proided by kernel to disable and enable bottom halves respactively.

Process Scheduling

1) what is process scheduling ?

Process of selecting which process to give chance to run is called Process scheduling. It is responsibility of processor to decide which process to run, when and for how long. For multiprogramming system scheduler is base . Scheduler has to decide which process to run and which process has to wait. Scheduler has to allot the time to each process efficiently and make the efficient use of Processor so that user can feel that all the processes is executing at the same time. Read more about process scheduling??

http://www.tutorialsdaddy.com/2015/09/process-scheduling/

2) What is cooperative multitasking and pre-emptive multitasking?

Co-operative multitasking:-In this method process does not stop running until it voluntarily stops. The act of process voluntarily stopping itself called yielding.

Preemptive multitasking:-In this kind of multitasking scheduler decides which process to start running and which process to stop. Good thing in this

multitasking is that scheduler can take the global decision. All the linux kind OS comes in preemptive scheduling flavor.

3) What is yielding?

The act of process voluntarily stopping itself called yielding.

4) What is limitation of cooperative multitasking?

- Scheduler cannot take globalize?? decision.
- Any process can monopolize the system.

5) I/O bound versus Processor bound process?

I/O bound processes spend most of the time waiting for some input/output event to occur and take very less time of processor.Processor bound process takes most of the time executing the code . These kind of process runs continuously without waiting for any input/output event and stopped while printed by compiler.

6) What is process priority?

Process priority is a way of specifying the process importance in the system.

7) What kind of priority is maintained in linux?

Linux implements two types of process priority value.

1) Nice value:-A number from -20 to +19 with a default value of 0. Large nice value corresponds to less priority.
2) Real Time Priority:-A value from 0 to 99. Higher the real time value means higher priority.

8)**What is nice value?**

Read answer of previous question.

9) What is virtual run time?

The Vruntime variable stores the virtual runtime of a process ehich is the actual runtime (the amount of time spent running) normalized by the number of runnable processor. CFS uses Vruntime to account for how long a process has run and thus how longer it ought to run.

10) What are the available scheduling classes in linux?

- Completely Fair Schedule class
- RT schedule class

11) Which type of scheduling used in linux?

Go through Process scheduling.

http://www.tutorialsdaddy.com/2015/09/process-scheduling/

12) How next task is picked for scheduling ?

CFS (complete fair scheduler)search the leftmost child of RB tree which has smallest Vruntime. It is performed by pick_next_entity(). If it returns NULL , it means there is no task to run so schedules the idle task.

13) what is scheduler entry point in linux?

The main entry point in the process schedule is the function schedule() defined in kernel/sched.c .This function is called by rest of kernel to schedule another process.schedule() is a generic call and it internally calls each scheduler class and check highest priority task from highest priority scheduler class. It?? is done by pick_next_task().

14) what is waitqueus?

Its a simple list of process waiting for an event to occur. Waitqueue are represented in the kernel by wake_queue_head_t. Waitqueue are created statically via DECLARE_WAITQUEUE() or dynamically via init_waitqueue_head() . Read more about waitqueus??

15) How context switching is handled in linux?

Switching from one runnable process to another runnable process is called Context switching. In linux context switching is handled by context_switch() function. This function defined in /kernel/sched.c . When some process terminates then context_switch() is called by schedule() which does two basic things:-

1) switch_mm() :-This call is to switch the virtual memory mapping of currently running process to upcoming process.

2)switch_to():-This call is to switch the processor state from previous process to current process. It save the current info of current processor state register.

16) What is user preemption and kernel preemption?

User Preemption

When the kernel is about to return to user-space , need-reschedule is set and therefore the scheduler is invoked.

Kernel Preemption

Any task in the kernel which is not holding any lock can be preempted. In linux we maintain a variable preempt_count in the thread_info structure. When process acquired a lock the preempt_value is increased by 1 and vice versa. whenever this value is 0 we can preempt the kernel.

Syscalls

1) What is syscalls ?

System Calls provide an interface between hardware and user space. System calls are required when we have to use any?? service of kernel. Suppose we want to change the date and time of system in that case we need to request to kernel through Syscalls. Read more about Syscalls.

http://tutorialsdaddy.com/tutorials/syscalls.php?course=Linux%20Kernel

2) How system calls are implemented in linux ?

Go through Learn Syscalls.

http://tutorialsdaddy.com/tutorials/syscalls.php?course=Linux%20Kernel

3) What happens when process in userspace calls a syscall ?

Read Syscalls

http://tutorialsdaddy.com/tutorials/syscalls.php?course=Linux%20Kernel

4) What is the need of verifying parameter in definition of syscall ?

System call should check all the parameter that its valid and legal. Because syscall run in kernel space if user will give some invalid parameter then system security may suffer. For example we should check whether the process

descriptor or process PID etc. is valid or not. Every time whenever syscall request is coming syscall handler should check whether the process has the permission to access the requested resource or not.

5) What is system calls context ?

Execution of Linux kernel on behalf of A system calls is called system calls context.

6) Why it is not recommended to writing new syscall ?

Below are the cons of implementing syscalls because of which its not recommended to implement syscalls

- You need a syscall number, which needs to be officially assigned to you.
- After the system call is in a stable series kernel, it is written in stone.The interface cannot change without breaking user-space applications.
- Each architecture needs to separately register the system call and support it.
- System calls are not easily used from scripts and cannot be accessed directly fromthe filesystem.
- Because you need an assigned syscall number, it is hard to maintain and use a system call outside of the master kernel tree.
- For simple exchanges of information, a system call is overkill.