UNIT V CASE STUDY

1. Linux uses a time-sharing algorithm _____

a) to pair preemptive scheduling between multiple processes
b) for tasks where absolute priorities are more important than fairness
c) all of the mentioned
d) none of the mentioned
Answer: a
Explanation: None.

2. The first linux kernel which supports the SMP hardware?
a) linux 0.1
b) linux 1.0
c) linux 1.2
d) linux 2.0
Answer: d
Explanation: None.

3. Which one of the following linux file system does not support journaling feature?
a) ext2
b) ext3
c) ext4
d) none of the mentioned
Answer: a
Explanation: None.

4. Which binary format is supported by linux?

a) a.out
b) elf
c) both a.out and ELF
d) none of the mentioned
Answer: c
Explanation: None.
5. Which one of the following bootloader is not used by linux?
a) GRUB
b) LILO
c) NTLDR
d) None of the mentioned
Answer: c
Explanation: None.
6. The first process launched by the linux kernel is _____
a) init process
b) zombie process
c) batch process
d) boot process
Answer: a
Explanation: None.
7. Which desktop environment is not used in any linux distribution?
a) gnome
b) kde
c) unity
d) none of the mentioned
Answer: d
Explanation: None.

8. Standard set of functions through which interacts with kernel is defined by

_____

a) system libraries
b) kernel code
c) compilers
d) utility programs
Answer: a
Explanation: None.
9. What is Linux?
a) single user, single tasking
b) single user, multitasking
c) multi user, single tasking
d) multi user, multitasking
Answer: d
Explanation: None.
10. Which one of the following is not a linux distribution?
a) debian
b) gentoo
c) open SUSE
d) multics
Answer: d
Explanation: None.
1. Which one of the following is not shared by threads?
a) program counter
b) stack
c) both program counter and stack
d) none of the mentioned

Answer: c
Explanation: None.
2. A process can be _____
a) single threaded
b) multithreaded
c) both single threaded and multithreaded
d) none of the mentioned

Answer: c
Explanation: None.
3. If one thread opens a file with read privileges then _____
a) other threads in the another process can also read from that file
b) other threads in the same process can also read from that file
c) any other thread can not read from that file
d) all of the mentioned
Answer: b
Explanation: None.
4. The time required to create a new thread in an existing process is _____
a) greater than the time required to create a new process
b) less than the time required to create a new process
c) equal to the time required to create a new process
d) none of the mentioned
Answer: b

Explanation: None.
5. When the event for which a thread is blocked occurs?
a) thread moves to the ready queue
b) thread remains blocked
c) thread completes
d) a new thread is provided
Answer: a
Explanation: None.
6. The jacketing technique is used to _____
a) convert a blocking system call into non blocking system call
b) create a new thread
c) communicate between threads
d) terminate a thread
Answer: a
Explanation: None.
7. Termination of the process terminates _____
a) first thread of the process
b) first two threads of the process
c) all threads within the process
d) no thread within the process
Answer: c
Explanation: None.
8. Which one of the following is not a valid state of a thread?
a) running
b) parsing

c) ready
d) blocked
Answer: b
Explanation: None.
9. The register context and stacks of a thread are deallocated when the thread?
a) terminates
b) blocks
c) unblocks
d) spawns
Answer: a
Explanation: None.
10. Thread synchronization is required because _____
a) all threads of a process share the same address space
b) all threads of a process share the same global variables
c) all threads of a process can share the same files
d) all of the mentioned
Answer: d
Explanation: None.
1. A thread is also called _____
a) Light Weight Process(LWP)
b) Heavy Weight Process(HWP)

c) Process
d) None of the mentioned
Answer: a

Explanation: None.
2. A thread shares its resources(like data section, code section, open files, signals) with _____
a) other process similar to the one that the thread belongs to
b) other threads that belong to similar processes
c) other threads that belong to the same process
d) all of the mentioned
Answer: c
Explanation: None.
3. A heavy weight process _____
a) has multiple threads of execution
b) has a single thread of execution
c) can have multiple or a single thread for execution
d) none of the mentioned
Answer: b
Explanation: None.
4. A process having multiple threads of control implies _____
a) it can do more than one task at a time
b) it can do only one task at a time, but much faster
c) it has to use only one thread per process
d) none of the mentioned
Answer: a
Explanation: None.

5. Multithreading an interactive program will increase responsiveness to the user by _____

a) continuing to run even if a part of it is blocked
b) waiting for one part to finish before the other begins
c) asking the user to decide the order of multithreading
d) none of the mentioned
Answer: a
Explanation: None.

6. Resource sharing helps _____

a) share the memory and resources of the process to which the threads belong
b) an application have several different threads of activity all within the same address space
c) reduce the address space that a process could potentially use
d) all of the mentioned
Answer: d
Explanation: None.

7. Multithreading on a multi – CPU machine _____

a) decreases concurrency
b) increases concurrency
c) doesn’t affect the concurrency
d) can increase or decrease the concurrency
Answer: b

Explanation: None.
8. The kernel is _____ of user threads.
a) a part of
b) the creator of
c) unaware of
d) aware of
Answer: c
Explanation: None.
9. If the kernel is single threaded, then any user level thread performing a blocking system call will _____
a) cause the entire process to run along with the other threads
b) cause the thread to block with the other threads running
c) cause the entire process to block even if the

other threads are available to run
d) none of the mentioned
Answer: c
Explanation: None.
10. Because the kernel thread management is done by the Operating System itself _____
a) kernel threads are faster to create than user threads
b) kernel threads are slower to create than user threads
c) kernel threads are easier to manage as well as create then user threads

d) none of the mentioned
Answer: b
Explanation: None.
11. If a kernel thread performs a blocking system call, _____
a) the kernel can schedule another thread in the application for execution
b) the kernel cannot schedule another thread in the same application for execution
c) the kernel must schedule another thread of a different application for execution
d) the kernel must schedule another thread of the same application on a different processor
Answer: a
Explanation: None.
12. Which of the following is FALSE?
a) Context switch time is longer for kernel level threads than for user level threads
b) User level threads do not need any hardware support
c) Related kernel level threads can be scheduled on different processors in a multiprocessor system
d) Blocking one kernel level thread blocks all other related threads
Answer: d
Explanation: None.
1. The model in which one kernel thread is mapped to many user-level threads is called _____

a) Many to One model
b) One to Many model
c) Many to Many model
d) One to One model
Answer: a
Explanation: None.
2. The model in which one user-level thread is mapped to many kernel level threads is called _____
a) Many to One model
b) One to Many model
c) Many to Many model
d) One to One model
Answer: b
Explanation: None.
3. In the Many to One model, if a thread makes a blocking system call _____
a) the entire process will be blocked
b) a part of the process will stay blocked, with the rest running
c) the entire process will run
d) none of the mentioned
Answer: a
Explanation: None.
4. In the Many to One model, multiple threads are unable to run in parallel on multiprocessors because of _____
a) only one thread can access the kernel at a time
b) many user threads have access to just one

kernel thread
c) there is only one kernel thread
d) none of the mentioned
Answer: a
Explanation: None.
5. The One to One model allows _____

a) increased concurrency
b) decreased concurrency
c) increased or decreased concurrency
d) concurrency equivalent to other models
Answer: a
Explanation: None.
6. In the One to One model when a thread makes a blocking system call _____
a) other threads are strictly prohibited from running
b) other threads are allowed to run
c) other threads only from other processes are allowed to run
d) none of the mentioned
Answer: b
Explanation: None.
7. Which of the following is the drawback of the One to One Model?
a) increased concurrency provided by this model
b) decreased concurrency provided by this model

c) creating so many threads at once can crash the system
d) creating a user thread requires creating the corresponding kernel thread
Answer: d
Explanation: None.
8. When is the Many to One model at an advantage?
a) When the program does not need multithreading
b) When the program has to be multi-threaded
c) When there is a single processor
d) None of the mentioned
Answer: a
Explanation: None.
9. In the Many to Many model true concurrency cannot be gained because _____
a) the kernel can schedule only one thread at a time
b) there are too many threads to handle
c) it is hard to map threads with each other
d) none of the mentioned
Answer: a
Explanation: None.
10. In the Many to Many models when a thread performs a blocking system call _____
a) other threads are strictly prohibited from

running
b) other threads are allowed to run
c) other threads only from other processes are allowed to run
d) none of the mentioned
Answer: b
Explanation: None.
1. Which of the following system calls does not return control to the calling point, on termination?
a) fork
b) exec
c) ioctl
d) longjmp
Answer: b
Explanation: None.
2. The following program results in the creation of?

```
main()
{
if(fork()>0)
sleep(100);
}
```

a) an orphan process
b) a zombie process

c) a process that executes forever
d) none of the mentioned
Answer: b
Explanation: None.

3. Which of the following system calls transforms executable binary file into a process?
a) fork
b) exec
c) ioctl
d) longjmp
Answer: b
Explanation: None.
4. How many times the following C program prints yes?

```
 main()
 {
 fork();fork();printf("yes");
 }
```

a) only once
b) twice
c) four times
d) eight times
Answer: c
Explanation: None.
5. Which of the following calls never returns an error?
a) getpid
b) fork
c) ioctl
d) open
Answer: a
Explanation: None.
6. A fork system call will fail if

_____
a) the previously executed statement is also a fork call
b) the limit on the maximum number of processes in the system would be executed
c) the limit on the minimum number of processes that can be under execution by a single user would be executed
d) all of the mentioned
Answer: b
Explanation: None.
7. If a thread invokes the exec system call

_____
a) only the exec executes as a separate process
b) the program specified in the parameter to exec will replace the entire process
c) the exec is ignored as it is invoked by a thread
d) none of the mentioned
Answer: b
Explanation: None.
8. If exec is called immediately after forking

_____
a) the program specified in the parameter to exec will replace the entire process
b) all the threads will be duplicated
c) all the threads may be duplicated
d) none of the mentioned
Answer: a

Explanation: None.
9. If a process does not call exec after forking
_____
a) the program specified in the parameter to exec will replace the entire process
b) all the threads should be duplicated
c) all the threads should not be duplicated
d) none of the mentioned
Answer: b
Explanation: The new process is purely based on fork, due to no exec command, duplication will be done.

1. What is Thread cancellation?
a) the task of destroying the thread once its work is done
b) the task of removing a thread once its work is done
c) the task of terminating a thread before it has completed
d) none of the mentioned
Answer: c
Explanation: None.
2. When a web page is loading, and the user presses a button on the browser to stop loading the page?
a) the thread loading the page continues with the loading
b) the thread loading the page does not stop but continues with another task

c) the thread loading the page is paused
d) the thread loading the page is cancelled
Answer: d
Explanation: None.
3. When one thread immediately terminates the target thread, it is called _____
a) Asynchronous cancellation
b) Systematic cancellation
c) Sudden Termination
d) Deferred cancellation
Answer: a
Explanation: None.
4. When the target thread periodically checks if it should terminate and terminates itself in an orderly manner, it is called?
a) Asynchronous cancellation
b) Systematic cancellation
c) Sudden Termination
d) Deferred cancellation
Answer: d
Explanation: None.
5. Cancelling a thread asynchronously _____
a) frees all the resources properly
b) may not free each resource
c) spoils the process execution
d) none of the mentioned
Answer: b
Explanation: None.
6. Cancellation point is the point where

_____
a) the thread can be cancelled – safely or otherwise doesn’t matter
b) the thread can be cancelled safely
c) the whole process can be cancelled safely
d) none of the mentioned
Answer: b
Explanation: None.
7. If multiple threads are concurrently searching through a database and one thread returns the result then the remaining threads must be _____
a) continued
b) cancelled
c) protected
d) none of the mentioned
Answer: b
Explanation: None.
1. Signals that occur at the same time, are presented to the process _____
a) one at a time, in a particular order
b) one at a time, in no particular order
c) all at a time
d) none of the mentioned
Answer: b
Explanation: None.
2. Which of the following is not TRUE?
a) Processes may send each other signals
b) Kernel may send signals internally
c) A field is updated in the signal table when

the signal is sent
d) Each signal is maintained by a single bit

Answer: c
Explanation: A field is updated in the process table when the signal is sent.
3. Signals of a given type _____
a) are queued
b) are all sent as one
c) cannot be queued
d) none of the mentioned
Answer: b
Explanation: The signal handler will be invoked only once.
4. The three ways in which a process responds to a signal are _____
a) ignoring the signal
b) handling the signal
c) performing some default action
d) all of the mentioned
Answer: d
Explanation: None.
5. Signals are identified by _____
a) signal identifiers
b) signal handlers
c) signal actions
d) none of the mentioned
Answer: a
Explanation: None.
6. When a process blocks the receipt of

certain signals?
a) The signals are delivered
b) The signals are not delivered
c) The signals are received until they are unblocked
d) The signals are received by the process once they are delivered
Answer: a
Explanation: None.
7. The _____ maintains pending and blocked bit vectors in the context of each process.
a) CPU
b) Memory
c) Process
d) Kernel
Answer: d
Explanation: None.
8. In UNIX, the set of masked signals can be set or cleared using the _____ function.
a) sigmask
b) sigmaskproc
c) sigprocmask
d) sigproc
Answer: c
Explanation: None.
9. The usefulness of signals as a general inter process communication mechanism is limited because _____
a) they do not work between processes

b) they are user generated
c) they cannot carry information directly
d) none of the mentioned
Answer: c
Explanation: None.
10. The usual effect of abnormal termination of a program is _____
a) core dump file generation
b) system crash
c) program switch
d) signal destruction
Answer: a
Explanation: None.
11. In UNIX, the abort() function sends the _____ signal to the calling process, causing abnormal termination.
a) SIGTERM
b) SIGSTOP
c) SIGABORT
d) SIGABRT

Answer: d
Explanation: None.
12. In most cases, if a process is sent a signal while it is executing a system call _____
a) the system call will continue execution and the signal will be ignored completely
b) the system call is interrupted by the signal, and the signal handler comes in

c) the signal has no effect until the system call completes
d) none of the mentioned
Answer: c
Explanation: None.
13. A process can never be sure that a signal it has sent _____
a) has which identifier
b) has not been lost
c) has been sent
d) all of the mentioned
Answer: b
Explanation: None.
14. In UNIX, the _____ system call is used to send a signal.
a) sig
b) send
c) kill
d) sigsend
Answer: c
Explanation: None.
1. Thread pools are useful when _____
a) when we need to limit the number of threads running in the application at the same time
b) when we need to limit the number of threads running in the application as a whole
c) when we need to arrange the ordering of threads

d) none of the mentioned
Answer: a
Explanation: None.
2. Instead of starting a new thread for every task to execute concurrently, the task can be passed to a _____
a) process
b) thread pool
c) thread queue
d) none of the mentioned
Answer: b
Explanation: None.
3. Each connection arriving at multi threaded servers via network is generally _____
a) is directly put into the blocking queue
b) is wrapped as a task and passed on to a thread pool
c) is kept in a normal queue and then sent to the blocking queue from where it is dequeued
d) none of the mentioned
Answer: b
Explanation: None.
4. What is the idea behind thread pools?
a) a number of threads are created at process startup and placed in a pool where they sit and wait for work
b) when a process begins, a pool of threads is chosen from the many existing and each thread is allotted equal amount of work

c) all threads in a pool distribute the task equally among themselves
d) none of the mentioned
Answer: a
Explanation: None.
5. If the thread pool contains no available thread _____
a) the server runs a new process
b) the server goes to another thread pool
c) the server demands for a new pool creation
d) the server waits until one becomes free

Answer: d
Explanation: None.
6. Thread pools help in _____
a) servicing multiple requests using one thread
b) servicing a single request using multiple threads from the pool
c) faster servicing of requests with an existing thread rather than waiting to create a new thread
d) none of the mentioned
Answer: c
Explanation: None.
7. Thread pools limit the number of threads that exist at any one point, hence _____
a) not letting the system resources like CPU time and memory exhaust

b) helping a limited number of processes at a time
c) not serving all requests and ignoring many
d) none of the mentioned
Answer: a
Explanation: None.
8. The number of the threads in the pool can be decided on factors such as _____
a) number of CPUs in the system
b) amount of physical memory
c) expected number of concurrent client requests
d) all of the mentioned
Answer: d
Explanation: None.