Unix Threads Interview Questions And Answers Guide.

Global Guideline.

https://globalguideline.com/

Unix Threads Job Interview Preparation Guide.

Question # 1

What are the different kinds of threads?

Answer:-

Only two typs of thereads,that are user space and kernel supported threads.The user space threads can run only one machine.where as the kernel supported can run on two or more machincs simulatinously.

Read More Answers.

Question # 2

Explain What are the main families of threads?

Answer:-

There are two main families of threads:
POSIX-style threads, which generally run on Unix systems.
Microsoft-style threads, which generally run on PCs.
These families can be further subdivided.

Read More Answers.

Question # 3

Explain similarities between thread and process?

Answer:-

- each process must have at least one thread running within it, and each thread must be running within a process.
- each process gets it own address space and memory allocation by OS, where as thread makes use of its parent process resources.
- when parent process dies, all its child process dies, but v-v is not true

Read More Answers.

Question # 4

When should we use thread-safe "_r" library calls?

Answer:-

If your system provides threads, it will probably provide a set of thread-safe variants of standard C library routines. A small number of these are mandated by the POSIX standard, and many Unix vendors provide their own useful supersets, including functions such as gethostbyname_r(). Unfortunately, the supersets that different vendors support do not necessarily overlap, so you can only safely use the standard POSIX-mandated functions. The thread-safe routines are conceptually "cleaner" than their stateful counterparts, though, so it is good practice to use them wherever and whenever you can.

Read More Answers.

Question # 5

What are the Performance differences between User-space threads and Kernel-supported threads.?

Answer:-

In terms of context switch time, user-space threads are the fastest, with two-level threads coming next (all other things being equal). However, if you have a multiprocessor, user-level threads can only be run on a single CPU, while both two-level and pure kernel-supported threads can be run on multiple CPUs simultaneously.

Read More Answers.

Question # 6

What are the Architectural differences between User-space threads, and Kernel-supported threads?

Answer:-

User-space threads live without any support from the kernel; they maintain all of their state in user space.
Since the kernel does not know about them, they cannot be scheduled to run on multiple processors in parallel.
Kernel-supported threads fall into two classes.
In a "pure" kernel-supported system, the kernel is responsible for scheduling all threads.
Systems in which the kernel cooperates with a user-level library to do scheduling are known as two-level, or hybrid, systems. Typically, the kernel schedules LWPs, and the user-level library schedules threads onto LWPs.
Because of its performance problems (caused by the need to cross the user/kernel protection boundary twice for every thread context switch), the former class has fewer members than does the latter (at least on Unix variants). Both classes allow threads to be run across multiple processors in parallel.

Read More Answers.

Question # 7

What is Scheduling?

Answer:-

Scheduling is a key concept in computer multitasking, multiprocessing operating system and real-time operating system designs. Scheduling refers to the way processes are assigned to run on the available CPUs, since there are typically many more processes running than there are available CPUs. This assignment is carried out by softwares known as a scheduler and dispatcher.
The scheduler is concerned mainly with:
 * Throughput - number of processes that complete their execution per time unit.
 * Latency, specifically

Read More Answers.

Question # 8

What is Protection boundary?

Answer:-

A protection boundary protects one software subsystem on a computer from another, in such a way that only data that is explicitly shared across such a boundary is accessible to the entities on both sides. In general, all code within a protection boundary will have access to all data within that boundary.
The canonical example of a protection boundary on most modern systems is that between processes and the kernel. The kernel is protected from processes, so that they can only examine or change its internal state in certain strictly-defined ways.
Protection boundaries also exist between individual processes on most modern systems. This prevents one buggy or malicious process from wreaking havoc on others.

Read More Answers.

Question # 9

How to edit the network interface device type in container(zone) in Solaris 10?

Answer:-

just edit
vi /etc/zones/<zonename.xml> file
then change the value of physical=<nic>
and save it.
after that reboot zone.
by
zoneadm -z <zonename> boot
zlogin <zonename>

Read More Answers.

Question # 10

Explain What are threads?

Answer:-

A thread is an encapsulation of the flow of control in a program. Most people are used to writing single-threaded programs - that is, programs that only execute one path through their code "at a time". Multithreaded programs may have several threads running through different code paths "simultaneously".

Read More Answers.

Question # 11

How to work UNIX commands on Windows XP without installing UNIX O/S in PC?

Answer:-

Install virtual machine for running unix command that may be available from different different vendor like Ex:- vmware player,etc

Read More Answers.

Question # 12

Explain Critical section?

Answer:-

Necessary and sufficient conditions for a solution to the c.s. problem:

1. Mutual Exclusion --- if is executing in one of its critical sections, no , , is executing in its critical sections.
2. Progress --- a process operating outside of its critical section cannot prevent other processes from entering theirs; processes attempting to enter their critical sections simultaneously must decide which process enters eventually.
3. Bounded Waiting --- a process attempting to enter its critical region will be able to do so eventually.

Assumptions:

1. No assumptions made about relative speed of processes
2. No process may remain in its critical section indefinitely (may not terminate in its critical section)
3. A memory operation (read or write) is atomic --- cannot be interrupted. For now, we do not assume indivisible RMW cycles.

Read More Answers.

OS Unix Most Popular Interview Topics.

1 : Unix Inter-Process Communication (IPC) Frequently Asked Interview Questions and Answers Guide.

2 : Unix System Calls Frequently Asked Interview Questions and Answers Guide.

3 : Unix General Frequently Asked Interview Questions and Answers Guide.

About Global Guideline.

Global Guideline is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of **Interview Questions with Answers** and much more. Learn the most common technologies at Global Guideline. We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

* This PDF was generated from https://GlobalGuideline.com at **November 29th, 2023**

* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and e-mail us we will fix it.

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers

Follow us on Twitter for latest Jobs and interview preparation guides
https://twitter.com/InterviewGuide

Best Of Luck.

Global Guideline Team
https://GlobalGuideline.com
Info@globalguideline.com