Aidan Conlin
Lab 9: System Calls
CPE 2600 121
12/10/2024

Part 1: Information Maintenance System Calls

1. **clock_gettime**: Retrieves the current time, including seconds and nanoseconds, from specified clocks like CLOCK_REALTIME or CLOCK_MONOTONIC. It is used for timing and synchronization.
2. **uname**: Provides basic system information such as the OS name, release version, and hardware type. It's often used for diagnostics and reporting system details.
3. **gethostname**: Fetches the system's hostname, which is used to identify the device on a network. The hostname is stored in a user-provided buffer.
4. **get_nprocs**: Returns the number of CPUs available on the system, to try and optimize parallel processing or resource management. It only counts CPUs that are currently online.
5. **sysconf**: Queries runtime system configuration values, such as the number of processors or memory page size and adapts to the system's current state. It is essential for system-specific program behavior.
6. **getpagesize**: Retrieves the size of a memory page in bytes, which is critical for memory allocation and virtual memory management. Page size depends on the system architecture.

Part 2: Process Control System Calls

1. **getpid:** Retrieves the process ID of the calling process, used for identifying, debugging, or signaling a specific process.
2. **getpriority**: Gets the scheduling priority of a process, process group, or user, which ranges from -20 (highest) to 19 (lowest) and is used for performance tuning.
3. **sched_getscheduler**: Returns the scheduling policy (normal, FIFO, round-robin) of a process, enabling analysis or optimization of scheduling behavior.

4. **getcpu**: Retrieves the CPU and NUMA node where the thread is running, used for optimizing performance in multi-core or NUMA systems.
5. **getrusage**: Provides detailed resource usage metrics (CPU time, memory, I/O) for profiling and monitoring processes or threads.
6. **getrlimit**: Fetches resource limits (like max file size or CPU time) for a process, ensuring it operates within system constraints.
7. **nice**: Adjusts the scheduling priority of a process, increasing or decreasing its "niceness" value, where higher values reduce priority, enabling better control of resource allocation.
8. **nanosleep**: Suspends the execution of a process for a specified time with nanosecond precision, useful for implementing delays or timing-sensitive operations