OVERHEAD

Peter J. Denning, Naval Postgraduate School, Monterey, California

January 2008
Rev 5/26/08

Abstract: Overhead is the cost of general functions of an operating system that cannot be conveniently associated with any user task.

Keywords: overhead, operating system, accounting

Overhead in computer systems is like overhead in organizations -- shared functions that benefit everyone but that cannot be conveniently associated with any one activity. In organizations, rent, furnishings, electricity, telephones, utilities, supplies, auditing, accounting, general clerical support, and management are counted as overhead. In computer systems, allocation of resources, scheduling, conflict resolution, error correction, exceptional conditions, protection, security, performance monitoring, auditing, accounting, input/output control, caching, distributed functions, and network protocols are all counted as overhead. As in organizations, excessive overhead diminishes capacity and increases cost without increasing productivity. Overhead in computer systems manifests as slower processing, less memory, less network bandwidth, or bigger latencies than expected.

Overhead is not always easy to measure. The time an operating system spends in supervisor state is not pure overhead because many important operations requested by user tasks are implemented as system functions that run in supervisor state, for example, input/output, file operations, and message-passing. A measurement that an operating system spends 80% of its time in supervisor state does not mean that the system spends only 20% of its time doing useful work: We need to know what portion of the 80% is spent responding to requests from user tasks. Moreover, many operating systems use special coprocessors to perform important overhead tasks such as authentication, virtual memory control, external communications, or peripheral device management; these coprocessors do not diminish processor capacity, memory capacity, or bandwidth available to user tasks. When coprocessors perform system functions, a measurement that a processor spends 90% of the time running user tasks does not mean that overhead is low.

Listed below are the main functions that usually count as overhead. Each one has a cost in processing time, memory space, network bandwidth, and latency (response time).

Allocation of Resources. Many resources such as CPU cycles, disk sectors, main-memory page frames, local-network packet slots, and shared files can be used by only one task at a time. To prevent conflicts and deadlocks, operating systems implement schedulers for these resources. The time spent running a scheduler and the memory occupied by a scheduler's queues count as overhead.

Error Correction. Data are stored and transmitted with redundant bits that permit detecting and correcting errors. These bits consume some space and bandwidth.

Exceptional Conditions. Most system functions have normal and error returns; the instructions that test for and respond to errors consume some space and processing time. Examples are arithmetic contingencies, data transmission failures, addressing snags, and illegal actions.

Protection and Security. Monitors, firewalls, authenticators, backup systems, virus detectors, and other means of securing systems against unauthorized use, denial of service, and intruders are necessary but often expensive.

Performance Monitoring, Auditing and Accounting. Recording key actions and events, logging each task's usage of resources, figuring costs and billings to users of the system, and generating statistics on resource usage and performance cost.

Input/Output Control. Many I/O operations are easy to specify at the user level -- for example, open or read a file. But the device spoolers and drivers can be quite complex because they must queue up requests from multiple tasks, translate each request into the low-level instruction sets of the devices, automatically work around known problems such as bad disk sectors, and handle interrupt conditions from their devices.

Caching. The speed of operations on secondary storage devices or remote servers can often be significantly improved by keeping a copy of the data in a local memory. Microcomputer register-windows, virtual memories, disk drivers, open-file managers, network browsers, and Internet edge servers are among the many prominent examples of caching. Caching consumes memory and processing time to locate and load copies of items into the cache and to maintain consistency with the originals.

Distributed Functions. Modern operating systems distribute their functions transparently over a collection of servers and workstations connected by a high-speed local network. Examples are file servers, printing servers, compute servers, authentication servers, and workstations. Maintaining the appearance that files, printers, processors, and login-sites are location independent significantly improves usability but is not cheap.

Network Protocols. Protocols for opening connections, transferring data, obtaining encryption keys, routing, and authenticating access all cost processing time, memory, and bandwidth.

Mainframe operating systems often charge users for processor, memory, and disk usage. In these systems, overhead will be charged back to users as percentage increases in each of these components. In personal computer networks and time-shared research computing systems, users are not charged for resource usage or overhead.

Overhead detracts from system performance only to the extent that the overhead functions do not add to the productivity of user tasks. Many services are provided by the system to relieve programmers from having to provide these functions themselves or to prevent expensive breakdowns. As long as the system can provide these functions more efficiently than its users, the resulting increases in overhead are offset by better service, improved performance, and lower overall costs.