A PARALLEL BETWEEN OPERATING SYSTEMS AND HUMAN GOVERNMENT

Andrew Klossner
Soothsayer Software
1428 Lake Boulevard
St Joseph, Michigan 49085

Operating systems and governments exist to perform analogous functions. A parallel can be drawn: operating systems are to computer programs as governments are to people. An understanding of the manifestations of this parallel should allow the computer engineer and the political scientist to draw from each other's field.

For purposes of comparison, let us make a number of loose analogies between the realms of government and operating system.

People or companies and programs (or processes) are the individuals within the system/state. Individually or in groups, they do the useful work.

The operating system or government is the authority in its sphere. The authority is responsible for maintaining order, and is otherwise charged with furthering the goals of the system/state as a whole, and of the separate individuals. When emphasis is placed upon the goals of the state as a whole, the government is socialist, or, in the extreme case, totalitarian; when emphasis is on the goals of separate individuals, as in the United States, the authority does not itself do useful work, but maintains an environment conducive to useful work by individuals.

Property can be defined as those resources which can be allocated for a great length of time. Examples of property include real estate and data files. The authority is the means by which the assignment of property to individuals is protected.

The authority is given reign over those tasks which must be done, but are better done centrally rather than individually. Maintenance of communications and transportation channels, such as telephone lines, public roads, and I/O busses are either done by the authority or assigned on a monopoly basis to an individual.

The authority defines and enforces rules of behavior. An individual who attacks the authority or another individual is dealt with by the authority's agents of enforcement. The authority also provides security against external attack, as by defensive armed forces or by password checking before granting entry.

The authority includes mechanisms to handle disputes between individuals. Contract law can be compared to the implementation of semaphores as a means that the authority provides to foster cooperation to the mutual individual benefit.

The authority handles relationships with any other system/states. A distributed network is like a community of nations; the various authorities are often very different and may not understand each other, and find it advantageous to interact using well defined protocols.

With these similarities between government and operating systems, we can proceed to examine philosophies of political science for ideas applicable to operating system design.

The authority by its nature does no useful work, but provides overhead services to individuals. To do this, the authority consumes the same resources as the individuals; an operating system requires CPU cycles and memory space; a government requires tax dollars. This tax is in proportion to the number of overhead services which the authority provides.

Among the heterogenous group of individuals will be some who greatly need overhead services and appreciate a large authority, and some who need very few services and would prefer a smaller authority which exacts less tax. There are various philosophies of government prescribing different optimal authority sizes, which are perhaps applicable to operating system design.

The "large government" train of thought believes that many non-essential yet important duties should reside with the government. Examples of such duties include postal service, local libraries, and municipal garbage collection. The "small government" school believes that the public sector should be charged with a minimum of such tasks, arguing that they can be provided by the private sector, thus allowing individuals to purchase only those services that they need and fostering competition which leads to improved service delivery. Proponents of large government argue that a company may be less concerned with delivery of quality service than with improving its own profit, and that many services, such as libraries, can not be made self-supporting, but are vital in fostering such social goals as a well informed public. Opponents respond that a government is inherently unsuited to the efficient delivery of a service and that enterprises which are not self-supporting are not worth maintaining. A further point is that services which are provided by the government are usually forbidden to competing private firms, and so any alternative to the government's service is unavailable.

These philosophies of government carry over to philosophies of operating system design. In the beginning, governments and operating systems provided only the very basic services: common defense, formulation and enforcement of the law, unit record and disk I/O. Then came an era of letting the authority do "everything." Out of the depression grew the New Deal and the welfare state; out of batch monitors grew OS/360. Later there arose a clamor for authority to be thoughtfully pruned and restructured. Politicians are elected with promises to trim bureaucracy and cut spending; new operating systems have less "bells and whistles" and provide a smoother set of services.

The same methods of analysis now being used to determine whether a service is best provided by government can be used to decide whether a feature should be supported by an operating system, or by a library subroutine or separate process. An example is support of screen manipulation commands such as erase screen, position cursor randomly, and draw window. The operating system can be augmented by primitives which would perform these functions correctly for several different types of terminal, or, alternatively, a set of subroutines could provide this support. If the operating system does it, each individual user program occupies less memory than if it had to load the subroutines; the functions will likely execute more quickly; and the functionality is provided through a well defined interface, since there will not be different versions of the subroutine library. On the other hand, the necessary code might occupy system memory, even when not needed; the routines are impossible for the user to modify; and screen support may get in the way of programs trying to do related obscure things with the terminal, such as reading from an auxiliary keypad.

When designing an operating system for a small computer, considerations of limited resources constitute cause to tighten system features. For example, although a tree structured file system has been shown to be extremely advantageous on medium and large scale computers, the increase in utility might not justify the memory space and number of disk I/O operations necessary on a microcomputer. Where necessary, a tree structured system can be implemented by the user with a set of standard routines, which would not need to be resident when a large program with no such requirement is run.

A case where computer science might bear on political science concerns the categorization of a government as socialist or laissez-faire according to whether the goals of the state or the goals of the individual are paramount. An operating system is inherently laissez-faire; it makes no attempt to prescribe what the user processes must do, but rather supports them in their efforts. An operating system which presumed to direct its subprocesses would be better described as a task oriented program, analogous to a company rather than to a government. The lesson would seem to be that a state functions better (or, at least, more like a computer) when the government is not itself concerned with production.

Conclusions. Because of parallels between governments and operating systems, political theories of government form and function can be useful to the operating system engineer. In particular, the question of which services should be provided by the operating system and which should be left to user-level subroutines is similar to the consideration of which tasks should be borne by a government rather than provided by private industry.