УДК 811.111:004.45

O. Medvedev, E. Rybaltovskaya

Fundamentals of operating systems

Belarusian National Technical University
Minsk, Belarus

Basically, computer is a device that accepts the message by the imputer and processes this message and stores the information at the storage devices and later gives an output of the message through the output devices.

Normally, a computer consists of a processing unit called the Central Processing Unit or the CPU and a form of memory.

Computational processes are abstract beings that inhabit computers. The evolution of a process is directed by a pattern of rules called a program. There is a natural classification of these processes into decidable, intermediate and complete.

A process is a running program. To implement virtualization of the central processing unit, and to implement it well, the operating system will need both some low-level machinery as well as some high-level intelligence.

One obvious component of machine state that comprises a process is its memory. Instructions lie in memory; the data that the running program reads and writes sits in memory as well. Thus the memory that the process can address (called its address space) is part of the process.

There are some ideas of what must be included in any interface of an operating system.

- Create: An operating system must include some method to create new processes.

- Destroy: As there is an interface for process creation, systems also provide an interface to destroy processes forcefully.
- Wait: Sometimes it is useful to wait for a process to stop running.
- Miscellaneous Control: Most operating systems provide some kind of method to suspend a+9 process and then resume it.
- Status: There are usually interfaces to get some status information about a process as well, such as how long it has run for, or what state it is in [1].

The operating system allocates the memory and gives it to the process. The operating system transfers control of the central processing unit to the newly-created process, and thus the program begins its execution. In a simplified view, a process can be in one of three states:

- Running: This means it is executing instructions.
- Ready: A process is ready to run but for some reason the operating has chosen not to run it at this given moment.
- Blocked: A process has performed some kind of operation that makes it not ready to run until some other event takes place.

The operating system is a program, and like any program, it has some key data structures that track various relevant pieces of information. To track the state of each process, the operating system likely will keep some kind of process list for all processes that are ready, as well as some additional information to track which process is currently running.

There are some other states a process can be in, beyond running, ready, and blocked. Sometimes a system will have an initial state that the process is in when it is being created. It could be placed in a final state where it has exited but has not yet been cleaned up. This final state allows other processes to examine the return code of the process and see if the just-finished process executed successfully.

Operating systems are replete with various important data structures. Sometimes people refer to the individual structure that stores information about a process as a Process Control Block (PCB).

As time sharing became more popular, new demands were placed on the operating system. In particular, allowing multiple programs to reside concurrently in memory makes protection an important issue. This abstraction is called the address space. Understanding this fundamental operating system abstraction of memory is the key to understanding how memory is virtualized [1].

The address space of a process contains all of the memory state of the running program. The program, while it is running, uses a stack to keep track of where it is in the function call chain as well as to allocate local variables and pass parameters and return values to and from routines.

Isolation is a key principle in building reliable systems. If two entities are properly isolated from one another, this implies that one can fail without affecting the other. Operating systems strive to isolate processes from each other and in this way prevent one from harming the other.

Computers have become the backbone of Information Technology and a major application in this sector is the Internet.

References:

1. Mode of access:
http://pages.cs.wisc.edu/~remzi/OSTEP/. – Date of access: 15.03.2016.