Virtual Process Controlling

Juri, Koval
Shevchenko National University of Kyiv
Kyiv, Ukraine
e-mail: smith@uis.kiev.ua

ABSTRACT
This paper provides the description of mechanism for virtual process controlling. The set of functions for such controlling is proposed and discussed.

Keywords
Virtual process, process controlling.

1. INTRODUCTION
Usual process in operating system [1] may be controlled. In UNIX operating system [2] the set of function to manage such control include next functions: fork(), exec(), kill(), nice(). In the next part of this paper such usual process may be mentioned as atomic. The virtual process, on contrast to atomic process, have no operating system to perform control over process. So the task of this paper is to determine the set and location of functions to perform control over virtual process. To solve this task let discover the role of each mentioned function in process controlling.

2. APPOINTMENT OF FUNCTIONS FOR PROCESS CONTROLLING
The fork() function is the only way to create new atomic process in UNIX operating system. So this determine the role of this function in process controlling. This role is create a process.

The exec() function is perform the action of code changing. As a result, the role of this function is code modification. It must be noted, that for UNIX and UNIX-like operating systems other methods of code modification are prohibited.

The kill() function perform the messaging or communicating task and finishing task. So, two roles are determined for this function: delete and communicate.

The nice() function is changing the activity of process. Non-UNIX operating systems prefer the notion of priority, but it is clear that nice is nice. The role of this function is to determine the processor resource availability.

So, five roles of functions are identified: create, delete, communicate, modify, and setting of activeness. Determining of function to perform such roles for virtual process is the task of next section.

3. SET OF FUNCTION TO CONTROL VIRTUAL PROCESS
3.1. Creating of virtual process
As stated in [3] virtual process can be started as usual atomic process. But it is very useful idea of fork() function to split process in two. So, split() function produce new virtual process like fork(). Other function that must be present for virtual process is function like usual function call. It is obvious that for compiling programming system there is no such function. But for interpreting the is eval() or call(0 functions. This functions produce a new part of process. As for virtual process the function run() will do the same.

3.2. Code modification of virtual process
Code modification may be done in many ways, but the overloading is the simplest. So function overlay() do this for virtual process.

3.3. Communication inside virtual process
Signal mechanism that used for kill() function can lost information. For the beginning of UNIX age this was acceptable. But nowadays messaging subsystems are wery good. So function for communicate will be cast(). This is more shorter then message. Also broadcast() can be used for broadcast communicating.

3.4. Finishing of virtual process
For internal virtual process finishing it is not enough to exit from process like to atomic one. The are many atomic process that create one virtual. So internal finishing of virtual process is like external finishing of process by kill(-9) for example. Nevertheless the name finite() is more better then terminate or kill.

3.5. Controlling of virtual process activeness
There is no special function for such controlling of virtual process. The activeness of virtual process is a number of atomic processes that are parts of the virtual process. So creating new atomic processes makes process faster or reducing the number of atomic processes makes it slower.

3.6. Joining of virtual processes
New possibility exists for virtual process is joining. It is possible as every virtual process uses separate name space. As a result joining of such spaces produce virtual process joining.

4. CONCLUSION
The next roles for virtual process controlling proposed: creating, code modification, communicating, finishing, and joining. Functions for each role are proposed.

REFERENCES
[1] A. Silberschatz, P. B. Galvin, and G. Gagne, OPERATING SYSTEM CONCEPTS with JAVA, 6th ed., JOHN WILEY & SONS, INC., USA, p. 1251, 2004
[2] K. Thompson, D. M. Ritchie, UNIX PROGRAMMER'S MANUAL, Bell Labs, USA, p. 194, November 1971, https://www.bell-labs.com/usr/dmr/www/1stEdman.html
[3] Iu. V. Krak, Iu. V. Koval, and A. B. Stavrovskyi, Virtual process: definition and application for gestures interface system creation, Bulletin of Taras Shevchenko National University of Kyiv, Series Physics & Mathematics, vol. 1, pp. 141-144, 2015