Operating Systems

List 3

Exercise 1

Describe an address space layout, base upon UNIX-like systems. Define process image term. Usually the first few megabytes of an address space are inaccessible to a user (programmer), why is so? How is the security of an application put at risk if code and data segments are always located at the same address? Can you shortly explain difficulties related to loading of dynamic libraries into into an address space?

Exercise 2

Provide a list of resources stored in a process control block. Why do they have to be associated with a process? Is it safe to store PCB partially in a user address space? What benefits can such placement provide?

Exercise 3

Give a detailed description of five state process model. Explain which transitions of that automata are triggered by an OS or by the process itself? What kind of actions initiate such transitions? Divide OS initiated actions into two groups - one controlled by short-term scheduler and the other - by long-term scheduler.

Exercise 4

Extend the five state process model to support memory swapping. Explain why OS could be interested in moving processes' memory partially or completely to the secondary memory? Given new states, please answer questions analogous to those presented in exercise no. 3.

Exercise 5

Describe in details the process of launching dynamically linked program from disk. Please base upon UNIX-like systems. Make sure you don't skip any important, but perhaps obvious step (eg. address space creation, resource allocation and management, dynamic linker tasks). How does process creation differ between Linux and Windows NT? Consider pros and cons of both solutions.

Exercise 6

Define process switching. How is such a task carried out? Give a list of reasons for which a process can be interrupted (ie. execution enters kernel-mode). How can it be utilized by a programmer? Give a list of events that can trigger process termination, try to generalize and organize them into groups. Are there such events, that normally would be recognized as a process failure, but could be potentially handled by the process or an OS to implement certain functionality? If yes, then give an example.

Exercise 7

Name three models of kernel mode execution. Describe shortly differences between them - consider advantages and disadvantages of each solution (focus on efficiency, security, process-kernel communication). For each model try to identify an OS which implements it.

Exercise 8

Describe changes to the thread state model compared to the five state process model. Why thread switching is usually much faster that process switching? Consider the same question in multiprocessor systems context. Define thread affinity term. How such a mechanism could benefit multithreaded applications?

Exercise 9

Describe differences between kernel level threads (KLTs) and user level threads (ULTs). Give a list of advantages and disadvantages of KLTs over ULTs and vice versa. How does an ULT library have to compensate for lack of kernel support? Define jacketing term. Consider hybrid threading model and show that it can fuse advantages of both models (ie. KLTs and ULTs).

Exercise 10

Let’s consider kernel level threads. There’re numerous resources assigned to a process. Which of these are shared between threads and which have to be private for a thread? Circumstantiate your answer. Find information about thread local storage and describe this mechanism.