ENPM695 – Secure Operating Systems

Homework – 4

Author – Syed Mohammad Ibrahim

UMD ID: iamibi

UID: 118428369

Email: iamibi@umd.edu

1. What is the difference between a process and a thread? What is the difference between user level thread and kernel level threads? (10 points total; 5 points each)

A. A process is a program that was launched from the ready state of the process cycle and is scheduled to be executed in the CPU. A process can create sub-processes called Child Process. A process has the following states associated with them: New, Ready, Running, Waiting, Terminated, and Suspended.

A thread on the other hand is a part of the process. That means, a process can contain multiple threads that build up the process. Threads are not isolated from other threads unlike a process. A thread has the following states associated with them: Running, Ready and Blocked.

The differences between a user level thread and a kernel level thread are:

User Level Thread	Kernel Level Thread
It is implemented by Users	It is implemented by the Operating System
Implementation is easy	Implementation is complicated
Context Switch requires no hardware support	Context Switch requires hardware support
Examples include Java Threads, POSIX threads	Examples include Window Solaris

2. Describe the process of starting up a program. Be as detailed as possible. (5 points)

A. Following steps are involved while a program is initiated:

- The user requests a program to be executed. This is usually achieved by clicking the binary file of the program or running the execution command on terminal.

- The operating system determines the name of the program.

- The operating system finds all the associated files and data stored on the secondary storage device.

- The operating system finds a memory block in the primary memory that is sufficient to accommodate the application run time data.

- The operating system makes a copy of the program and its data to this allocated block.

- The operating system starts provisioning the resources including any hardware device that might be required by the program to execute.

- At last, the program is started by the operating system and the state of program changes to Running.

3. What is a memory fence? How does it differ from base/bounds registers? (10 points)

A. A memory fence is a class of instructions which makes sure that the memory read or write instructions happen in the order the user expects. It is a concept implemented by hardware devices and high-level languages usually implement the Mutex and Semaphores at the lowest level using them.

A base-bound register is associated with each segment of data or code and defines the position in physical memory of word zero for that segment, the so-called base, and the number of words available to that segment, the so-called bound or limit. Whenever a process attempts to access the memory segment, the hardware of the system checks that the address of the word lies within the range 0 ← word address < bound and then adds the address to the value contained in the base register to give the physical address. It is a hardware used for Virtual Memory Allocation.

4. List the algorithms used to allocate memory to processes and describe each one. (15 points – 3 points each)

A. Following are the algorithms used for allocating memory to a process:

- First Fit
  In this algorithm, the operating system allocates the first free memory partition or hole that is large enough to accommodate the process. The algorithm program exits as soon as the first suitable partition is found.

- Best fit
  It deals with allocating the smallest free hole which meets the requirement of the process. The algorithm first traverses the whole list of free partitions and then picks up the smallest partition that is adequate. It then goes on to find the closest partition that is close to the actual process size needed.

- Worst fit
  This algorithm as the name suggests, searches for the largest available free partition so that the remaining partition left will be big enough to be useful. An opposite approach to the Best Fit algorithm.

5. List the six methods that can isolate processes from each other and the operating system. Provide 2 characteristics for each method. (15 points total – 2.5 points per method).

A. Methods to isolate processes from each other and operating system are:

- Fence
    - Memory is strictly divided into two separate areas by a specific memory location (or fence) namely, Very Restrictive and Predefined an inflexible space reserved for operating system.
    - Fence register: a technique used to provide an arbitrary and adjustable fenced area.
- Relocation
    - The process of taking a program written as if it began at address 0 and changing all addresses to reflect the actual address at which the program is in memory
    - A fence register can be used as an additional utility by using it as a hardware relocation device
- Base/Bounds Registers
    - Fence register gives a starting memory address (a base), but an upper limit address is added as well. It is known as boundary register
    - All memory allocation is forced to happen between the base register and the boundary register and appropriate checks are put in place to verify this rule is respected
- Tagged Architecture
    - A tagged architecture involves adding additional bits to every word of machine memory to help in identifying the access rights to that word
    - The bits can only be set by a privileged instructions like an operating system
- Segmentation
    - Involves dividing a program into separate pieces. Each piece has a logical unity, exhibiting a relationship among all its code or data values.
    - It was developed as a feasible means to produce the effect of the equivalent of an unbounded number of base/bounds registers
- Paging
    - The program is divided into equal size pieces referred to as pages with memory being divided into equal size units referred to as page frames
    - The object is made up of two parts, page and offset

6. List five (5) components of UNIX/Windows processes (10 points)

A. The five components of a process on UNIX/Windows are:

- Stack – Function and local variables allocation of memory
- Heap – Dynamic allocation of memory
- BSS – Uninitialized data
- Data – Initialized data
- Text – Source code and other related information

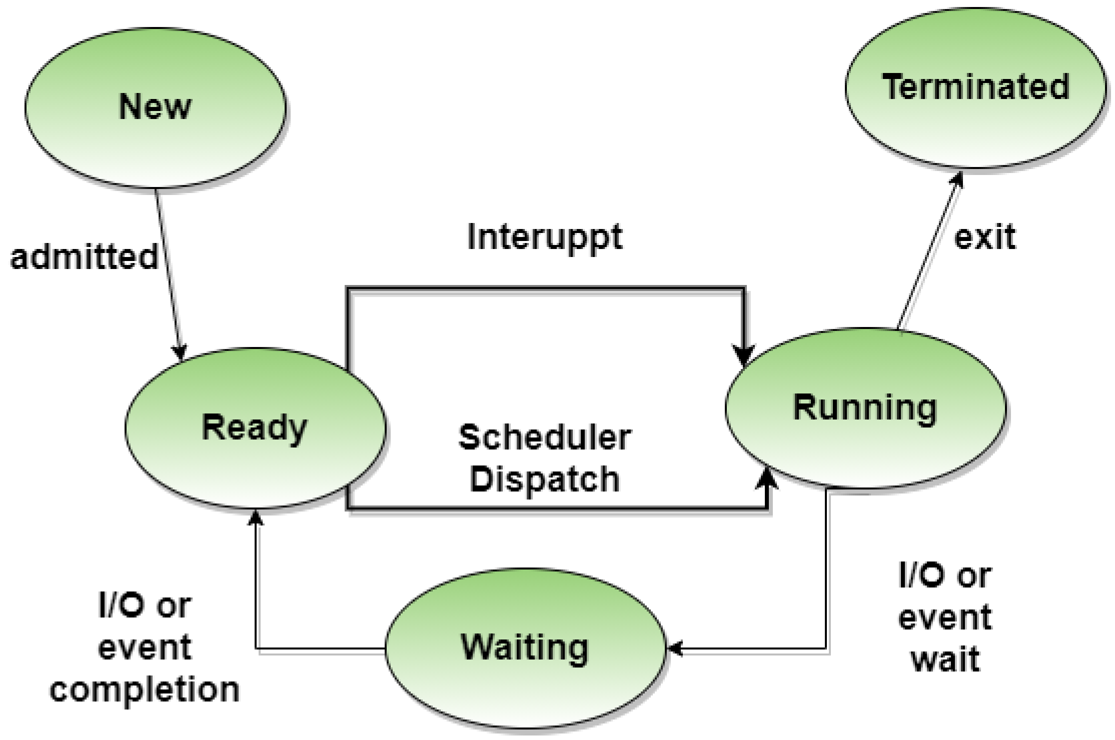
7. What is the difference between the two alternative methods used to keep track of free memory? (10 points)

A. Bit List involves a collection of bits where each bit corresponds to a disk block. It can contain two values, 0 and 1. A 0 indicates that the block is allocated and 1 as free. Finding the first free block is efficient as it requires scanning words or group of 8 bits in a bit list to find a non-zero word. The operations are fast.

Free list comprises of a chain of blocks containing free spaces around the memory. Each list contains a pointer to the next free block available. The start block number is stored in a cache or secondary memory for reference. The I/O is slow as it requires free list traversal.

8. Describe the process state transition for a generic process. List the various states as well as the transitions between the states. Where is the process state kept? (15 points)

A. A process generally has the following states



The individual components can be defined as:

- NEW- The process is being created.
- READY- The process is waiting to be assigned to a processor.
- RUNNING- Instructions are being executed.
- WAITING- The process is waiting for some event to occur (such as an I/O completion or reception of a signal).
- TERMINATED- The process has finished execution.

The steps involved for transition from one state to another are:

- Whenever a new process is created, it is admitted into ready state.

- If no other process is present at running state, it is dispatched to running based on scheduler dispatcher.
- If any higher priority process is ready, the uncompleted process will be sent to the waiting state from the running state.
- Whenever I/O or event is completed, the process will send back to ready state based on the interrupt signal given by the running state.
- Whenever the execution of a process is completed in running state, it will exit to terminate state, which is the completion of process.

There is a Process Control Block for each process, enclosing all the information about the process. It is also known as the task control block. It is a data structure, which holds the process state along with other process related information.