CHAPTER 2: PROCESSES

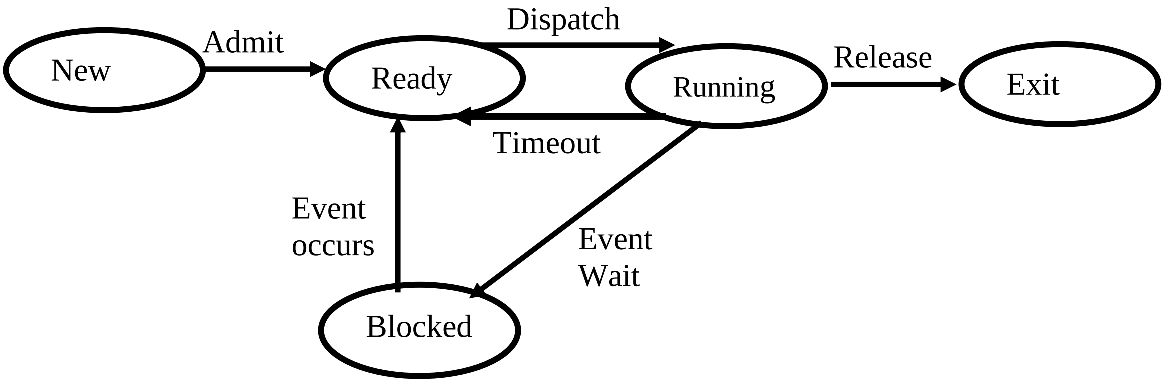
ANSWERS

1. What are the differences between multiprogramming and time-sharing?

   `Multiprogramming - When one job needs to wait for I/O, the processor can switch to the other job`

   `Time Sharing – multiple users simultaneously access the system with time quantum`

2. Draw the complete five State Process Model.

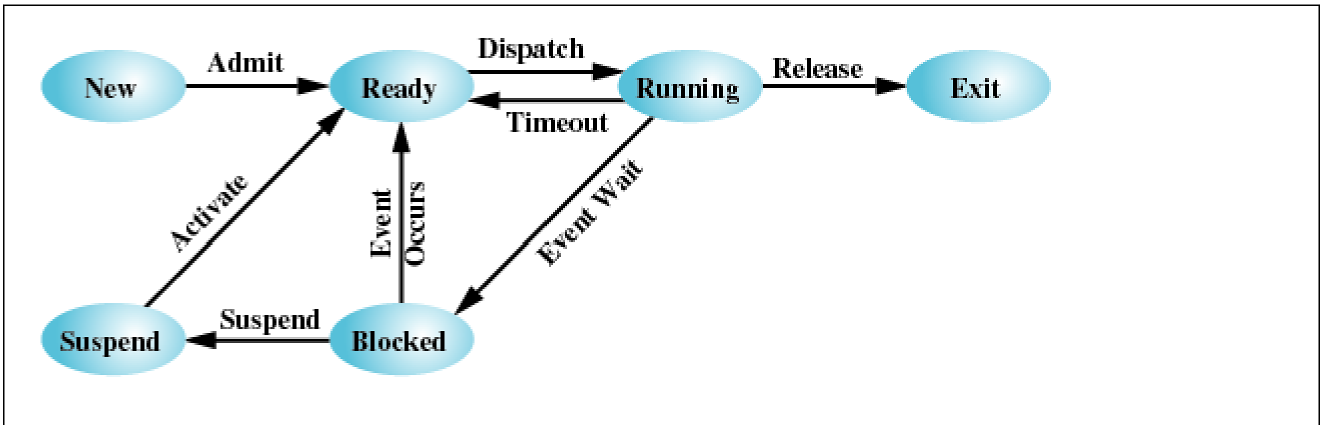


3. What is a suspended process? Give and explain two (2) suspended states.

   `Processes waiting for I/O are swapped to disk to free up more memory.`

   a. `Blocked/Suspend – a process in secondary memory and awaiting an event.`
   b. `Ready/Suspend - a process in secondary memory but available for execute as soon as it is loaded into main memory.`

4. Draw the Process State Transition Diagram with one Suspend states.
   Lukiskan gambar rajah peralihan keadaan proses dengan satu keadaan Suspend.



(a) With One Suspend State

5. Give <u>TWO</u> (2) examples of events that cause a state transition for a process from RUNNING to READY.
Berikan <u>DUA</u> (2) contoh peristiwa yang menyebabkan pertukaran keadaan bagi proses daripada RUNNING kepada READY.

Answer/Jawapan:

RUNNING----READY:

- The most commonly known situation is that currently running process has taken its share of time for execution (Time out).
- Also, in some events a process may have to be admitted from running to ready if a high priority process has occurred.

6. What is a process? Explain how operating system does to create a new process.
Apakah ia satu proses. Jelaskan bagaimanakah yang dilakukan oleh sistem pengoperasian untuk mencipta suatu proses yang baru.

A program in execution.
An instance of a program running in a computer.
The entity that can be assigned to and executed on a processor.
A unit of activity characterized by the execution of a sequence of instructions, a current state and associated set of system resources. (Any answer acceptable)

To create a new process, it can proceed as follows:
a. Assign a unique process identifier to the new process.
b. Allocate space for the process.
c. Initialize the process control block.
d. Set the appropriate linkages.
e. Create or expand other data structures.

7. List three general categories of information in a process control block.
Senaraikan tiga kategori-kategori am bagi maklumat yang terkandung dalam blok kawalan proses.

Process identification

Processor state information.
Process control information.

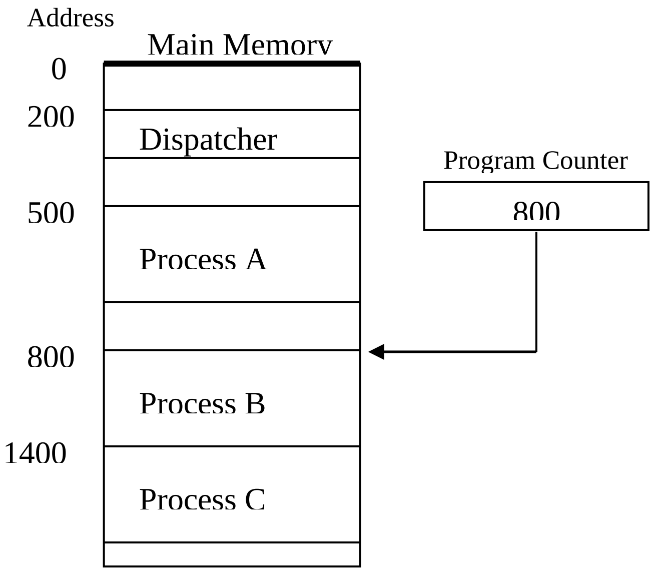
8. Give and explain five states of process model.
Beri dan terangkan mengenai model lima keadaan proses.

New - A process that just been created but has not yet been admitted to the pool of executable processes by the operating system
Ready - A process that is prepared to execute when given the opportunity
Running - The process that is currently being executed.
Blocked - A process that cannot execute until some event occurs, such as the completion of an I/O operation
Exit - A process that has been released from the pool of executable processes by the operating process

9. What are the steps performed by an operating system to create a new process?
Apakah langkah-langkah yang dilaksanakan oleh sistem pengoperasian untuk membentuk proses baru.

- Assign a unique process identifier
- Allocate space for the process
- Initialize process control block
- Set up appropriate linkages -Eg: add new process to linked list used for scheduling queue
- Create of expand other data structures -Eg: maintain an accounting file

10. Based on the information below, do the trace of processes. Assume Operating System allows a process to execute for maximum three instruction cycles, after which it is interrupted. Start with process A.
Berdasarkan maklumat di bawah, laksanakan pengesanan proses-proses. Andaikan sistem pengoperasian membenarkan proses melaksanakan tiga kitaran arahan, selepas itu ianya akan disampuk. Mulakan dengan proses A.



Process A → 5 instructions
Process B → 4 instructions, then request an I/O action
Process C → 7 instructions
Dispatcher → executes 2 instructions code

11. Give four types of relationship between thread and process implementation normally used with example of operating system using that method.
Berikan empat jenis perlaksanaan hubungan diantara bebenang dan prosses yang biasa digunakan beserta contoh sistem pengoperasi yang menggunakan kaedah tersebut.

- One process one thread eg. msdos
- One process multiple threads eg. Java run-time environment
- Multiple processes one thread per process eg. UNIX
- Multiple processes multiple threads per process eg. Solaris

12. How is the execution context of a process used by the operating system.
(Bagaimanakah 'execution context' bagi sesuatu proses digunakan oleh sistem pengoperasian).

- Contains the process elements
- Created and manage by the operating system
- Allows support for multiple processes

13. What is instruction trace?
(Apakah dia penjejak arahan?).

- Sequence of instruction that execute for a process
- Dispatcher switches the processor from one process to another

14. List four characteristics of a suspended process.
(Senaraikan empat ciri proses tergantung)

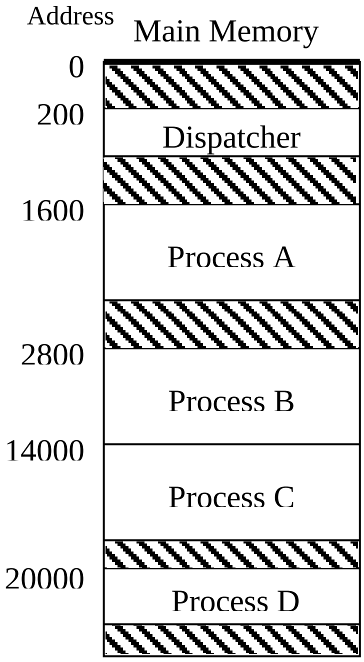
- Processor is faster than I/O so all processes could be waiting for I/O
- Swap these processes to disk to free up more memory
- Blocked state becomes suspend state when swapped to disk
- Two new states
    - Blocked/Suspend
    - ReadySuspend

15. Process Control Block contains specific information stored in a data structure. Describe four (4) elements of the PCB that characterize the process
Blok Pengawalan Proses (PCB) mengandungi maklumat tertentu yang disimpan di dalam struktur data. Terangkan empat (4) elemen PCB yang memperincikan sesuatu proses.

- Identifier; unique id to distinguish among process
- State; running, ready, suspend/block
- Priority level relative to other process
- Program counter; the address of the next instruction to be executed in main memory
- Memory Pointers

- Context data
- I/O status information

16. Consider the execution of four processes, each of them loaded into main memory as shown below:



Process A → 6 instructions
Process B → 4 instructions, then request an I/O action
Process C → 6 instructions
Process D → 5 instructions
Dispatcher → executes 3 instructions code

Assume the operating system allows a process to continue execution for a maximum of 3 instruction cycles, after which it is interrupted.

Show the **combined trace** of the four processes using the information given above.

Answer:

Trace of processes

Process A	Process B	Process C	Process D
1600 1601 1602 1603 1604 1605	2800 2801 2802 2803	14000 14001 14002 14003 14004 14005	20000 20001 20002 20003 20004

Combined trace:

1600
1601
1602
-------Time out
200
201
202
2800
2801
2802
-------Time out

200
201
202
20000
20001
20002
-------Time out
200
201
202
1603

200
201
202
14003
14004
14005
-------Terminate
200
201
202
20003

```
200                 1604                 20004
201                 1605                 -------Terminate
202                 -------Terminate
14000               200
14001               201
14002               202
-------Time out     2803
                    -------I/O request
```

Threads

17. List reasons why a mode switch between threads may be cheaper than a mode switch between processes. (Q 4.2)

    Answer: Less state information is involved.

18. Give four general examples of the use of threads in a single-user multiprocessing system.

19. Consider a multiprocessor system and a multithreaded program written using many-to-many threading model. Let the number of user-level threads in the program be more than the number of processors in the system. Discuss the performance implications of the following scenarios:

    a. The number of kernel threads allocated to the program is less than the number of processors.

    b. The number of kernel threads allocated to the program is equal to the number of processors.

    c. The number of kernel threads allocated to the program is greater than the number of processors but less than the number of user-level threads.

    (SGG 4.6)

    Answer:

    When a number of kernel threads is less than the number of processors, then some of the processors would remain idle since the scheduler maps only kernel threads to processors and not user-level threads to processors.

    When the number of kernel threads is exactly equal to the number of processors, then it is possible that all of the processors might be utilized simultaneously. However, when a kernel-thread blocks inside the kernel (due to a page fault or while invoking system calls), the corresponding processor would remain idle.

    When there are more kernel threads than processors, a blocked kernel thread could be swapped out in favour of another kernel thread that is ready to execute, thereby increasing the utilization of the multiprocessor system.

20. Give **TWO (2)** general examples of the use of threads in a single-user multiprocessing system.

Answer:

- Foreground/background work;
- asynchronous processing;
- speedup of execution by parallel processing of data;
- modular program structure.

21. Windows, Solaris, Linux, Mach, and OS/2 are the examples of operating systems that support multiple threads.
Windows, Solaris, Linux, Mach, dan OS/2 adalah contoh-contoh sistem pengoperasian yang menyokong pelbagai bebenang.

a. What is thread?
Apakah bebenang?

b. Explain the concept of multithreading.
Jelaskan konsep multi-bebenang.

c. Give TWO (2) benefits of using thread?
Berikan DUA (2) faedah menggunakan bebenang?

Answer:

a. Thread- lightweight process, a unit of execution

b. Multithreading: A program, which splits up into several threads, is said to be multithreading

Operating system supports multiple threads of execution within a single process

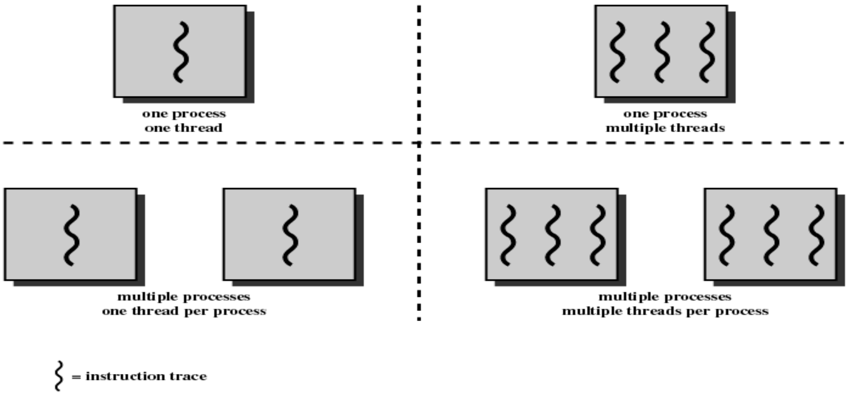


Figure 4.1 Threads and Processes [ANDE97]

c. Benefits of thread:
- Takes less time to create a new thread than a process
- Less time to terminate a thread than a process
- Less time to switch between two threads within the same process

- Since threads within the same process share memory and files, they can communicate with each other without invoking the kernel

22. What does a process do to start another process?

    It makes a system call that creates a new (child) process. Depending on the OS and the exact call used, this child process may be an entirely different program or it may be a copy of the parent process. This call might be a "fork," exec," "shell," or "clone" call, depending on the OS.

23. Distinguish between a process and a thread.

    A process is a program that is running. A thread is something less than a process. At any given point in the execution of a process we can save the processor context (registers and stack space) and start executing the process anew, perhaps at another point in the process. Each separate running (or waiting) processor context within a process is called a thread. Threads within a single process share address spaces and are thus less expensive to switch among than are processes.

24. Why do we usually say that kernel level threads are better than user level threads?

    The OS kernel is aware of kernel level threads, so that if one thread of a process makes a blocking OS call, the OS can dispatch another thread from that process with minimum context switching overhead. In order to accomplish such overlap with user level threads, the application must resort to asynchronous I/O and other OS calls - a much more complex task.

25. A thread is sometimes called as a lightweight process. The table below refers to two of the relationships between threads and processes.
    Bebenang juga dikenali sebagai pecahan proses atau proses ringan. Jadual di bawah merujuk kepada dua perhubungan antara bebenang dan proses.
    Describe each of the relationships below.
    Terangkan setiap perhubungan di bawah.

Threads : Process	**Description**
1 : 1	Each thread of execution is a unique process with its own address space and resources.
1 : M	A thread may migrate from one process environment to another. This allows a thread to be easily moved among distinct systems.

26. List <u>TWO</u> (2) benefits of threads over processes.
Senaraikan <u>DUA</u> (2) kelebihan bebenang berbanding proses.

- Takes less time to create a new thread than a process
- Less time to terminate a thread than a process
- Less time to switch between two threads within the same process
- Thread can communicate with each other without invoking the kernel

27. Give the differences between user-level threads and kernel-level threads.
Berikan perbezaan di antara bebenang tahap-pengguna dan bebenang tahap-kernel.

- ULT
  - All thread management is done by the application
  - The kernel is not aware of the existence of threads
- KLT
  - Kernel maintains context information for the process and the threads
  - Scheduling is done on a thread basis

28. Define multithreading.
Definisikan multi-bebenang.

Multithreading is a technique in which a process, executing an application, is divided into threads that can run concurrently.