University of Sargodha

BS 5th Term Exam 2019

Subject: Computer Science **Course: Operating System (CS:3621)**

Time Allowed: 2:30 Hours **Maximum Marks: 60**

Objective Part Compulsory

Q.No.1: Attempt all parts and each require answer 2 – 3 line (12*2=24)

i. What is Multiprogramming.

Multiprogramming is a rudimentary form of parallel processing in which several programs are run at the same time on a uniprocessor.

Multiprogramming is also the ability of an operating system to execute more than one program on a single processor machine. A computer running excel and Firefox browser simultaneously is an example of multiprogramming.

ii. Explain long term Scheduler.

It is also called a job scheduler. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.

iii. Under which circumstances do page faults occur.

A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid.

iv. Different types of Real-Time scheduling.

- Some commonly used RTOS scheduling algorithms are:
- Cooperative scheduling.
- Preemptive scheduling. Rate-monotonic scheduling. Round-robin scheduling. Fixed priority pre-emptive scheduling, an implementation of preemptive time slicing. ...
- Earliest Deadline First approach.
- Stochastic digraphs with multi-threaded graph traversal.

v. What are types of threads?

Thread is a single sequence stream within a process. Types are:

1. User Level thread (ULT)
2. Kernel Level Thread (KLT)

vi. What is a ready Queue?

This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.

vii. Differentiate between LFU algorithm and MFU algorithms.

LFU Algorithm: replaces page with smallest count; others were and will be used more. MFU Algorithm: based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

viii. What is the processor Affinity?

Processor affinity means you can specify which processor(s) a given process or thread should run on. Processor affinity, or CPU pinning, enables the binding and unbinding of a process or a thread to a central processing unit (CPU) or a range of CPUs, so that the process or thread will execute only on the designated CPU or CPUs rather than any CPU.

ix. What is Semaphores.

For the solution of the critical section problem one synchronization tool is used which is known as semaphores. A semaphore “S” is an integer value which is accessed through two standard operations such as wait and signal. There are two types of semaphores:

1. Binary semaphores
2. Counting semaphores

x. What is demand paging?

In virtual memory systems, demand paging is a type of swapping in which pages of data are not copied from disk to RAM until they are needed.

xi. What is spooling?

Spooling is an acronym for simultaneous peripheral operations on line. Spooling refers to putting data of various I/O jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices.

It is a kind of buffering mechanism or a process in which data is temporarily held to be used and executed by a device, program or the system. Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.

xii. What are turnaround time and response time?

Turnaround time is the amount of time elapsed from the time of submission to the time of completion whereas response time is the average time elapsed from submission until the first response is produced.

Subjective Part

Q2. Compare the memory organization schemes of contiguous memory allocation, pure segmentation and pure paging with respect to the following issue:

a. External fragmentation

Contiguous allocation with fixed-size partitions does not suffer from external fragmentation, but contiguous allocation with variable sized partitions does. Pure paging does not suffer from external fragmentation, since partitions and pages are fixed in size. Segmentation does suffer from external fragmentation.

b. Internal fragmentation

Segmentation and variable-sized partitions do not suffer from internal fragmentation, since by definition, a segment/partition is exactly as large as it needs to be. However, contiguous allocation with fixed-size partitions and paging both may suffer from internal fragmentation when partitions and pages are not completely filled.

c. Ability to share code across processes

- ✓ **Contiguous Allocation with Fixed-Size Partitions:** no support for code sharing across processes.
- ✓ **Contiguous Allocation with Variable-Size Partitions:** no support for code sharing across processes.
- ✓ **Pure Segmentation:** supports code sharing across processes. However, we must be careful to make sure that processes do not mix code and data in the same segment.
- ✓ **Pure Paging:** supports code sharing across processes. As in pure segmentation, we must make sure that processes do not mix code and data in the same page.

~~~~~~

Q3. a) Differentiate between Multiprocessors systems and Clustered Systems

Clustered system

- ✓ The clustered system usually consists of integrating several machines into one machine to complete tasks.
- ✓ Cluster systems are a mix of hardware clusters and software groups.
- ✓ Hardware clusters help to share high-performance disks among devices.
- ✓ The device clusters make both systems work together. Every node of the clustered systems contains the cluster program.

Multiprocessor system

- ✓ Multiprocessing refers to the ability of a computer system to accommodate more than one operation (program) at the very same time.
- ✓ Multiprocessing operating systems allow multiple programs can run simultaneously.
- ✓ Multiprocessor machines can be a single entity consisting of a number of CPUs.
- ✓ If two computers had to be purchased at a high pace, it would have to be modified almost every day in order to be available for service.

~~~~~~

b) Explain different types of Systems Calls.

Here are the types of system calls:

Process Control

These system calls deal with processes such as process creation, process termination etc.

File Management

These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.

Device Management

These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.

Information Maintenance

These system calls handle information and its transfer between the operating system and the user program.

Communication

These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

Some of the examples of all the above types of system calls in Windows and Unix are given as follows

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

~~~~~~

Q4. a) What is semaphore? How semaphore works for synchronizations of process?

Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows:

Wait

The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

```
wait(S)
{
  while (S<=0);

  S--;
}
```

Signal

The signal operation increments the value of its argument S.

```
signal(S)
{
  S++;
}
```

Types of Semaphores

There are two main types of semaphores i.e. counting semaphores and binary semaphores. Details about these are given as follows:

1. **Counting Semaphores**

These are integer value semaphores and have an unrestricted value domain. These semaphores are used to coordinate the resource access, where the semaphore count is the number of available resources. If the resources are added, semaphore count automatically incremented and if the resources are removed, the count is decremented.

2. **Binary Semaphores**

The binary semaphores are like counting semaphores but their value is restricted to 0 and 1. The wait operation only works when the semaphore is 1 and the signal operation succeeds when semaphore is 0. It is sometimes easier to implement binary semaphores than counting semaphores.

~~~~~~

b) Explain Process Creation and termination.

Process Creation

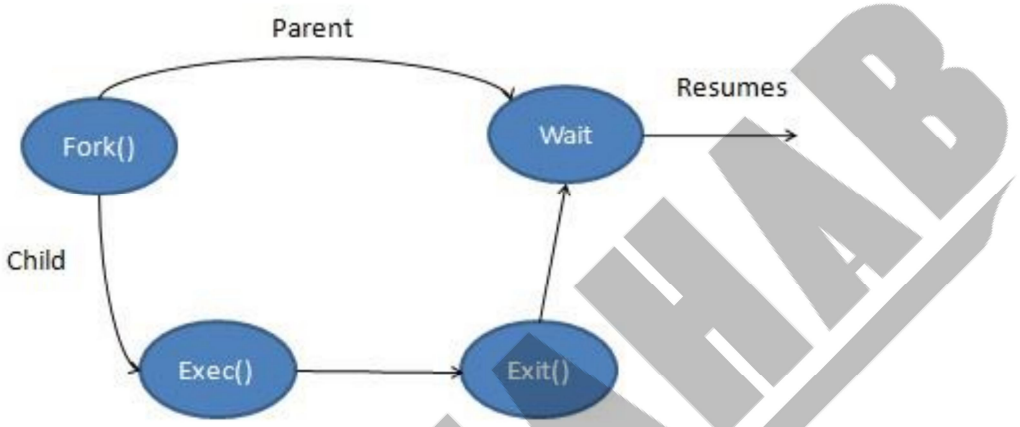
Process creation is achieved through the fork() system call. The newly created process is called the child process and the process that initiated it (or the process when execution is started) is called the parent process. After the fork() system call, now we have two processes - parent and child processes.

- Parent process create children processes, which, in turn create other processes, forming a tree of processes
- Generally, process identified and managed via a process identifier (pid)
- Resource sharing
  - ✓ Parent and children share all resources
  - ✓ Children share subset of parent's resources
  - ✓ Parent and child share no resources

- Execution
  - ✓ Parent and children execute concurrently
  - ✓ Parent waits until children terminate
- Address space
  - ✓ Child duplicate of parent
  - ✓ Child has a program loaded into it

UNIX examples

- ✓ fork system call creates new process
- ✓ exec system call used after a fork to replace the process' memory space with a new program



Process Termination

Process executes last statement and asks the operating system to delete it (exit)

- Output data from child to parent (via wait)
- Process' resources are deallocated by operating system

Parent may terminate execution of children processes (abort)

- Child has exceeded allocated resources
- Task assigned to child is no longer required
- If parent is exiting
  - ✓ Some operating systems do not allow child to continue if its parent terminates
    - ➢ All children terminated - cascading terminating

****~~~~~~****

Q5. Consider the following page reference: 4,3,1,2,1,1,1,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6. How many page faults would occur Optimal and Least Recently Used Page Replacement algorithms with (a) Five frames (a) Six frames. Remember that are initially empty.

Q6. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P1	10	2
P2	8	1
P3	2	5
P4	9	3
P5	4	4

The processes are assumed to arrive in the order P1, P2, P3, P4, P5 all at time 0.

a) **Draw four Gantt Charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a smaller priority number implies a higher priority) and RR (quantum =1)**
b) **Obtain the average waiting time.**