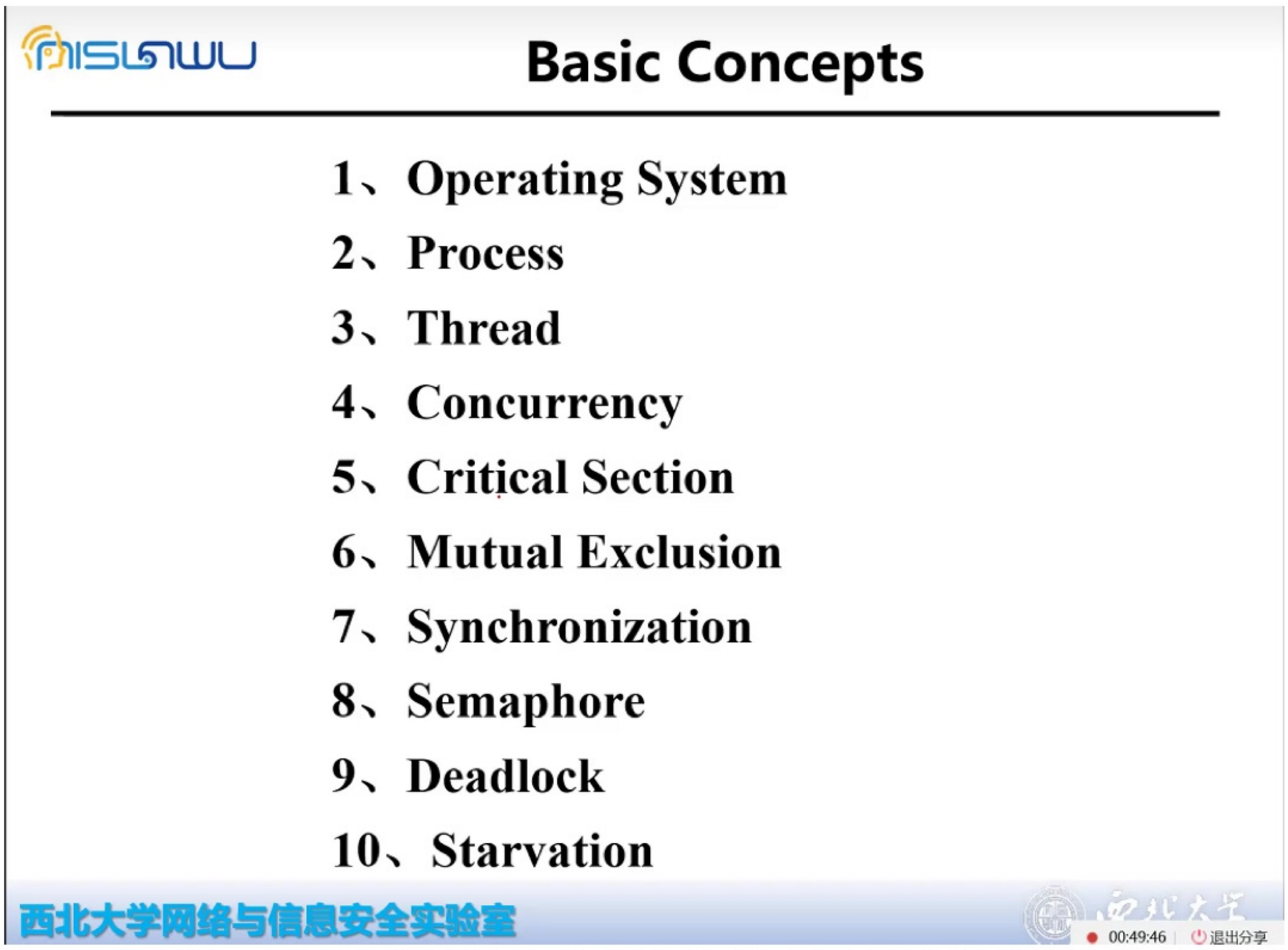

操作系统：负责管理协调硬件软件等计算机资源的工作，为上层用户，应用程序提供简单易用的服务，是一种系统软件

进程：是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位

线程：轻量级线程，线程是处理机调度的单位

并发：在一个时间段内操作系统存在多个程序，能够在同一个处理机上运行，但在任何一个时间点，只能有一个程序在运行，多个程序交替运行

临界区：

互斥：进程或者线程之间因相互竞争共享资源所产生的一种制约关系

同步：进程之间的一种协作关系，

信号量：是一个整数值，用于进程间信号的传递，用于阻塞进程或解除阻塞

死锁：进程的一种状态，是一组相互竞争系统资源的一种永久阻塞的状态
饥饿：因为资源竞争，导致长时间得不到处理器的一种状态

Fill In The Blank

1、The main objectives of the OS are: ( ), ( ) and ( )

2、( ) was introduced in modern operating systems, making concurrency and sharing possible.

3、When a processor time slice of a process is exhausted, the process should change to the ( ) state.

4、 When creating a new process, the first step is to assign a unique ( ) to the new process.

5、The first step in a full process switch is to save the ( ) of the process.

1. 便捷性、有效性、扩展能力
2. 多道程序处理
3. 就绪态
4. id
5. context:上下文

Fill In The Blank

6、In the instruction system, ( ) can only be used by the Operating System.

7、( ) is a special integer variable to be used for signaling among processes.

8、( ) is a special semaphore that takes on only the values 0 and 1.

9、The operating system's ( ) refers to its inherent flexibility in permitting functional modifications to the system without interfering with service.

6. 特权指令
7. 信号量
8. 二元信号量
9. 扩展能力

Single Choice

1. A process is in ( ) state when it is in the main memory and awaiting an event.

A) Ready B) Blocked/Suspend C) Blocked D) Ready/Suspend

2. Requested resources are granted to processes whenever possible with ( ).

A) deadlock detection B) deadlock avoidance

C) mutual exclusion D) preemption

抢占

C：阻塞态

A：死锁检测

Single Choice

3. In a three state Process Model, ( ) is not a valid state.

A. Block->Running B. Running->Block

C. Ready->Running D. Running->Ready

4) The permanent 永久 blocking of a set of processes that either compete 竞争 for system resources or communicate with each other is called: ( )

A. Starvation B. Deadlock

C. Prioritization D. Mutual exclusion

A

B

Single Choice

5. Suppose the initial value of a semaphore associated with a resource is 4 and the current value is -2. If V is the number of available resources and W is the number of the processes waiting for the resources, then the value of V and W are ( ) respectively.

A) 0, 2　　B) 4, 0　　C) 2, 0　　D) -2, 0

6. The end of a printout requested by a process will cause the process state to change from ( )

A. running -> ready　　B. running ->blocking

C. ready -> running　　D. blocking -> ready

西北大学网络与信息安全实验室

A

D

Single Choice

7. If there are five plotters in the system and all processes need to use two of them. Suppose each process is allowed to request only one at a time, then at most ( ) processes are allowed to compete without deadlock.

A 5　　B 2　　C 3　　D 4

8. In a concurrent system with multiple processes, there is certainly no deadlock due to contention ( ).

A printer　　B tape drive　　C CPU　　D disk

D

C

QUESTIONS

进程是资源分配的基本单位，线程是处理机调度的基本单位

1. What are the difference between process and thread?

线程切换时无需切换运行环境，代价小

2. Explain the relationship between program and process.

3. What are the difference between reusable resources and consumable resources?

可重用性资源：每一个可重用性资源中的单元只能分配给一个进程使用，不允许多个进程共享。进程在使用可重用性资源时，须按照这样的顺序：请求资源、使用资源、释放资源。系统中每一类可重用性资源中的单元数目是相对固定的，进程在运行期间既不能创建也不能删除它。

可消耗性资源：每一类可消耗性资源的单元数目在进程运行期间是可以不断变化的。有时它可以有许多，有时可能为0，进程在运行过程中，可以不断创造可消耗性资源的单元，将它们放入该资源类的缓冲区中，以增加该资源类的单元数目。进程在运行过程中，可以请求若干个可消耗性资源单元，用于进程自己的消耗，不再将它们返回给该资源类中。

程序和进程的关系：

1.程序是指令的有序 集合，其本身没有任何运行的含义，是一个静态的概念。而
进程是程序在处理机上的一次执行过程，它是一个动态的概念。

2.程序可以作为一种软件资料长期保存在某种介质上，而进程是有一定生命期的，
进程被创建后存在于 内存 中，进程消亡后生命期结束，不再存在。

3.程序的每次运行都将创建新的进程，而进程一旦消亡，就无法再被执行。

4.进程更能真实地描述并发，而程序不能(没有PCB)。

5.进程能够独立运行、独立分配资源和独立接受调度的基本单位，程序（没有 PCB）不能作为独立的单位运行。

QUESTIONS

4. What are the deadlock and livelock? Please briefly explain their difference?

5. List the requirements for mutual exclusion.

6. List reasons why a mode switch between threads may be cheaper than a mode switch between processes.

Problem Solving

1、Suppose current system states are shown as in the following table, at this time Available = (1,2,1).

Process	Claim			Allocation		
	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0
P2	6	3	1	5	1	1
P3	3	4	1	2	1	1
P4	4	2	2	0	2	0

Questions:

(1) When process P2 issues a resource request vector request2(1,1,0), can the system allocate the resource to it? why? If can, give a safety sequence;

(2) When process P2 issues a resource request vector request2(1,1,0), and then process P1 issues a resource request vector request1(0,1,0). Can the system allocate the resource to it? Why?

(3) When process P2 issues a resource request vector request2(1,1,0), and then process P3 issues a resource request vector request3(0,0,1). Can the system allocate the resource to it? Why?

Problem Solving

2. Suppose there are two processes P_1 and P_2 in an OS, they are concurrent executing with the same priority. The initial value of semaphore s_1 and s_2 are both zero. After the concurrent executing of P_1 and P_2, please give the value of x, y, and z. The programs are as follows:

```
P1 ( ) {                    P2( ){
  y=1;                        x=1;
  y=y+1;                      x=x+3;
  semSignal(s1);              semWait(s1);
  z=y+3;                      x=x+y;
  semWait(s2);                semSignal(s2);
  y=z+y;                      z=y+x;
}                           }
```

6 10 8

6 7 13