# 第四章

# 进程管理

方 钰



# 主要内容

- 4.1 UNIX时钟中断与系统调用
- 4.2 UNIX的进程调度状态
- 4.3 UNIX进程控制

# Process类

### Operating System



	名称	类型	含义
进程标识	p_uid	short	用户ID
	p_pid	int	进程标识数,进程编号
	p_ppid	int	父进程标识数
进程图象在内存中的位	p_addr	unsigned long	ppda区在物理内存中的起始地址
置信息	p_size	unsigned int	进程图象 (除代码段以外部分) 的长度,以字节单位
	p_textp	Text *	指向该进程所运行的代码段的描述符
进程调度相关信息	p_stat	ProcessState	进程当前的调度状态
	p_flag	int	进程标志位,可以将多个状态组合
	p_pri	int	进程优先数
	p_cpu	int	cpu值,用于计算p_pri
	p_nice	int	进程优先数微调参数
	p_time	int	进程在盘交换区上(或内存内)的驻留时间
	p_wchan	unsigned long	进程睡眠原因
信号与控制台终端	p_sig	int	进程信号
	p_ttyp	тту*	进程tty结构地址





	名称	类型		含义
进程标识	p_uid	short		用户ID
	p_pid	int		进程标识数,进程编号
	p_ppid	int		父进程标识数
进程图象在内存中的位 置信息	p_addr	unsigned lo	ng	ppda区在物理内存中的起始地址
	p_size	unsigned in	t	进程图象(除代码段以外部分)的长度,以字节单位
	p_textp	Text *		指向该进程所运行的代码段的描述符
进程调度相关信息	p_stat	ProcessStat	te	进程当前的调度状态
	p_flag	int	num Proc	essState
	p_pri	int {	{	
	p_cpu	int		: 0, /* 未初始化空状态
	p_nice	'SWAIT = 2, /*		
	p_time			
	n wchan	[~]		4,    /* 进程创建时的中间状态
信号与控制台终端	o_stat一定为 状态其中之			= 5,/*_进程终止时的中间状态 = 6

## Process类

#### **Operating System**



```
类型
                 名称
                                            含义
                                     4
                 p_uid
进程标识
                                                  SLOCK
                                                           SSYS
                          STWED
                                   STRC
                                          SSWAP
                                                                  SLOAD
                 p_pid
                          p_flag = SLOAD | SLOCK → p_flag=101; //二进制
                 p_ppid
                          p flag &= \simSLOCK; \rightarrow p flag=001;
                 p addr
进程图象在内存中的位
置信息
                 p_size
                          p flag &= \simSLOAD; \rightarrow p flag=000;
                                                            //二进制
                 p_textp
                          if (p flag & SLOAD) != 0)
                                           进程当前的调度状态
                 p_stat
                           ProcessState
进程调度相关信息
                 p flag
                                           进程标志位,可以将多个状态组合
                           int
                                           讲程优先数
                 n pri
                                                                  p flag可以是6个
             enum ProcessFlag /* 进程标志位 (用于进程图像换进换出)
                                                                  值的合理组合!
                            = 0x1, /* 进程图像在内存中
                SLOAD
               SSYS
                            = 0x2, /* 系统进程图像,不允许被换出
               SLOCK
                            = 0x4, /* 含有该标志的进程图像暂不允许换出
               SSWAP
                            = 0x8, /* 该进程被创建时图像就在交换区上
               STRC
信号与控制台终端
                            = 0x10, /* 父子进程跟踪标志, UNIX V6++未使用到
                                   /* 父子进程跟踪标志, UNIX V6++未使用到
                STWED
                            = 0x20
```

Tongji University, 2022

Fang Yu





	名称	类型	含义
进程标识	p_uid	short	用户ID
	p_pid	int	进程标识数,进程编号
	p_ppid	int	父进程标识数
进程图象在内存中的位 置信息	p_addr	unsigned long	ppda区在物理内存中的起始地址
	p_size	unsigned int	进程图象 (除代码段以外部分) 的长度, 以字节单位
	p_textp	Text *	指向该进程所运行的代码段的描述符
进程调度相关信息	p_stat	ProcessState	进程当前的调度状态
	p_flag	int	进程标志位,可以将多个状态组合
	p_pri	int	进程优先数 (值越大, 优先级越小)
	p_cpu	int	cpu值,用于计算p_pri
	p_nice	int	进程优先数微调参数
	p_time	int	进程在盘交换区上(或内存内)的驻留时间
	p_wchan	unsigned long	进程睡眠原因
信号与控制台终端	p_sig	int	进程信号
	p_ttyp	тту*	进程tty结构地址

# 进程优先数的计算方法

p\_pri = min { 255, ( p\_cpu / 16 + PUSER + p\_nice ) }

p\_pri的值越小,进程的优先级越高

PUSER: 固定偏置常数

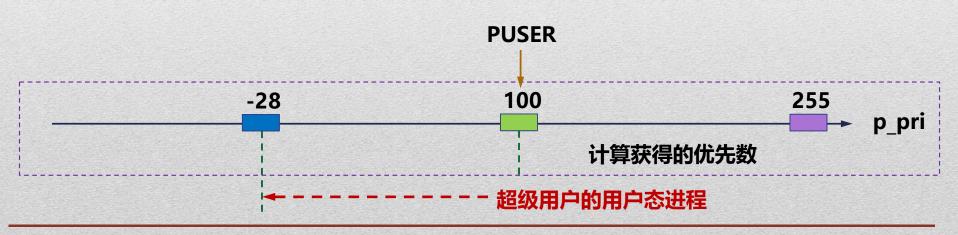
p nice: 相对优先程度

static const int PUSER = 100;

允许用户通过系统调用设置

超级用户: -128 ~ 20

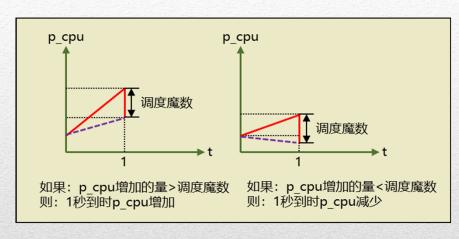
普通用户: 0~20



# 进程优先数的计算方法

p\_pri = min { 255, ( p\_cpu / 16 + PUSER + p\_nice ) }

p\_pri的值越小,进程的优先级越高



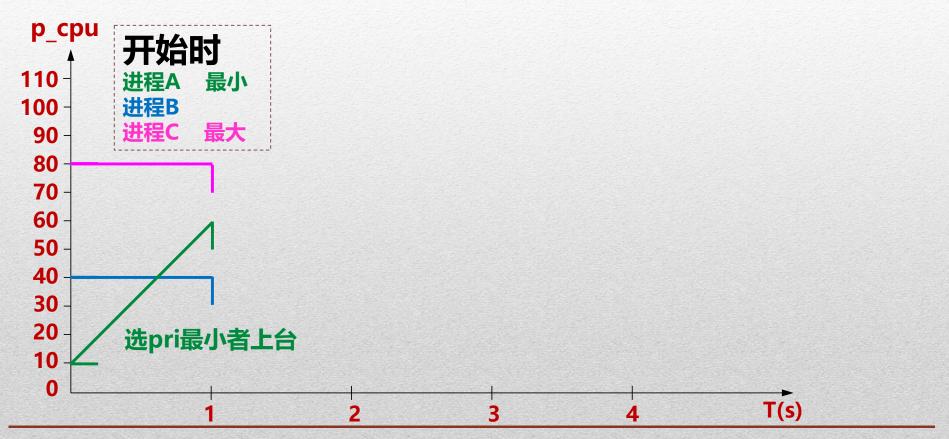
- 1. 连续占用处理机较长时间的进程: p cpu / , 优先数 / , 优先权 \
- 2. 较长时间内未使用处理机的进程:
  - p cpu ↘ ,优先数↘,优先权↗;
- 3. 虽频繁使用处理机,但每次时间很短的进程: p cpu ↘ , 优先数↘ , 优先权↗ 。

#### 防止高者恒高, 低者"饥饿"

# 进程优先数的计算方法

p\_pri = min { 255, ( p\_cpu / 16 + PUSER + p\_nice ) }

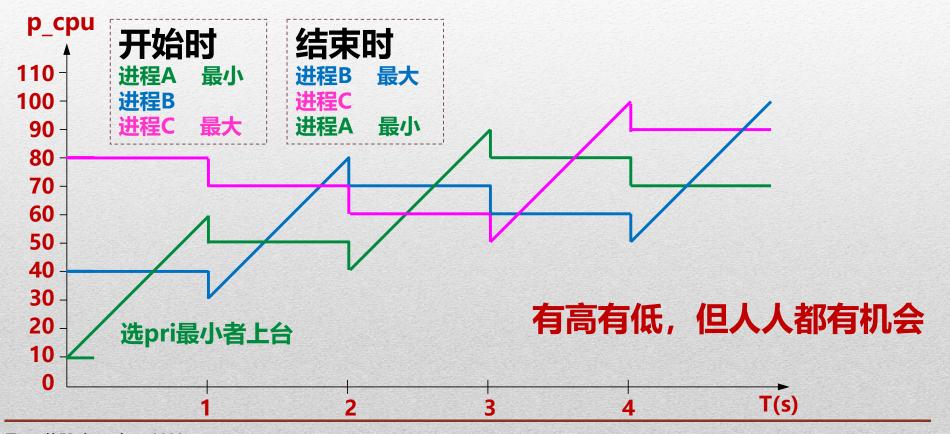
p\_pri的值越小,进程的优先级越高



# 进程优先数的计算方法

p\_pri = min { 255, ( p\_cpu / 16 + PUSER + p\_nice ) }

p\_pri的值越小,进程的优先级越高



Tongji University, 2022 Fang Yu

10

# Process类

### **Operating System**

6	1907	Prof
(OT)	PPP	
6	UNIN	2

	名称	类型	含义
进程标识	p_uid	short	用户ID
	p_pid	int	进程标识数,进程编号
	p_ppid	int	父进程标识数
进程图象在内存中的位	p_addr	unsigned long	ppda区在物理内存中的起始地址
置信息	p_size	unsigned int	进程图象 (除代码段以外部分) 的长度, 以字节单位
	p_textp	Text *	指向该进程所运行的代码段的描述符
进程调度相关信息	p_stat	ProcessState	进程当前的调度状态
	p_flag	int	进程标志位,可以将多个状态组合
	p_pri	int	进程优先数 (值越大, 优先级越小)
	p_cpu	int	cpu值,用于计算p_pri
	p_nice	int	进程优先数微调参数
	p_time	int	进程在盘交换区上(或内存内)的驻留时间
	p_wchan	unsigned long	进程睡眠原因
<b>1</b> 信号与控制台终端	p_sig	int	进程信号
	p_ttyp	тту*	进程tty结构地址



# 重新回看中断与系统调用的流程

一个<u>用户态</u>运行的进程PA 需要操作系统服务,如: 读写磁盘文件 使用只能通过 系统调用 INT 0x80 执行系统调用处理

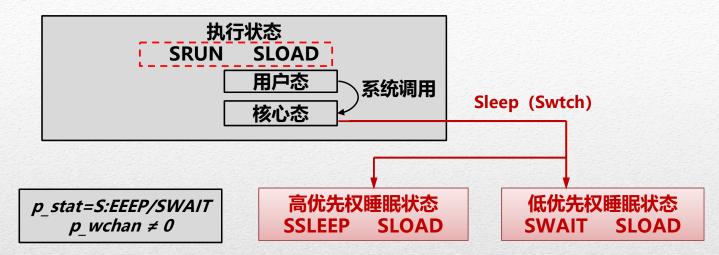


# UNIX进程的调度状态





# UNIX进程的调度状态



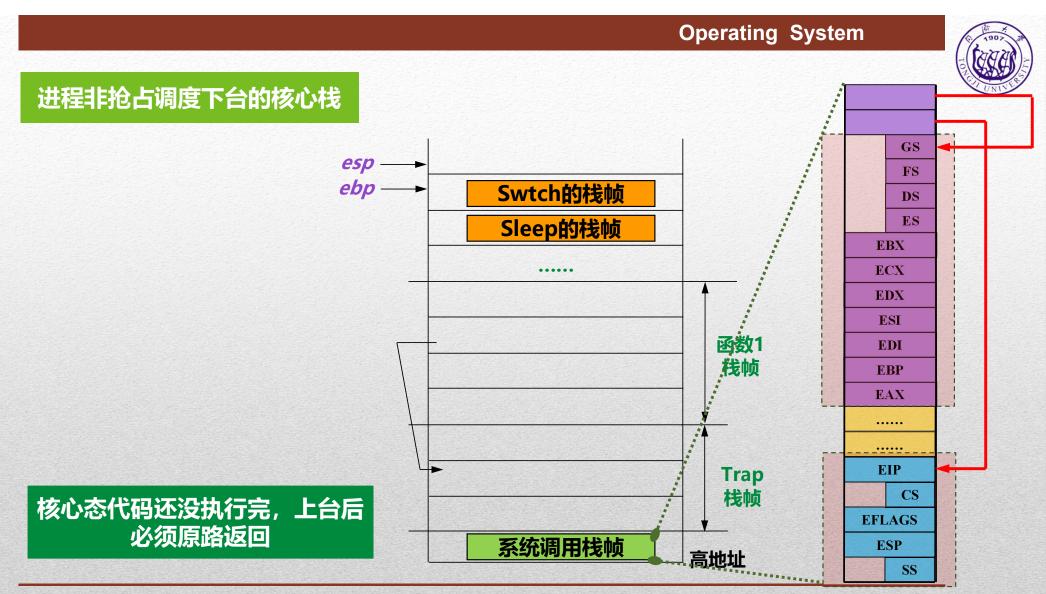


# 重新回看中断与系统调用的流程



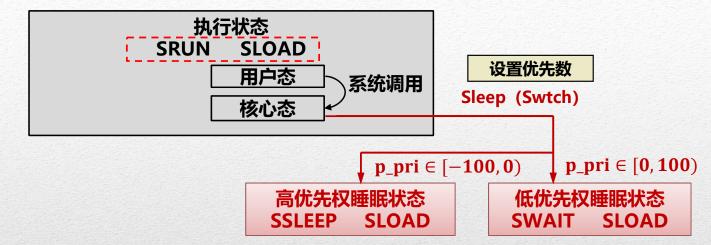
进程PA下台, 执行 → 阻塞 ② 新进程上台, 就绪 → 执行 ①

# 此时进程的核心栈是什么?



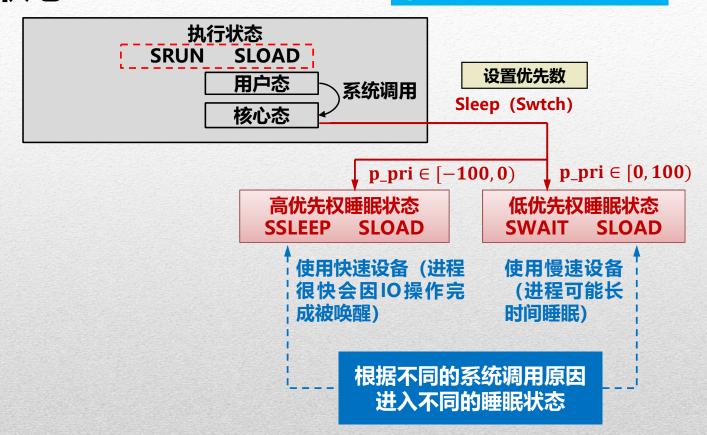
# UNIX进程的调度状态





# UNIX进程的调度状态

① 非抢占调度(设置优先数)



# UNIX进程的调度状态

#### ① 非抢占调度(设置优先数)



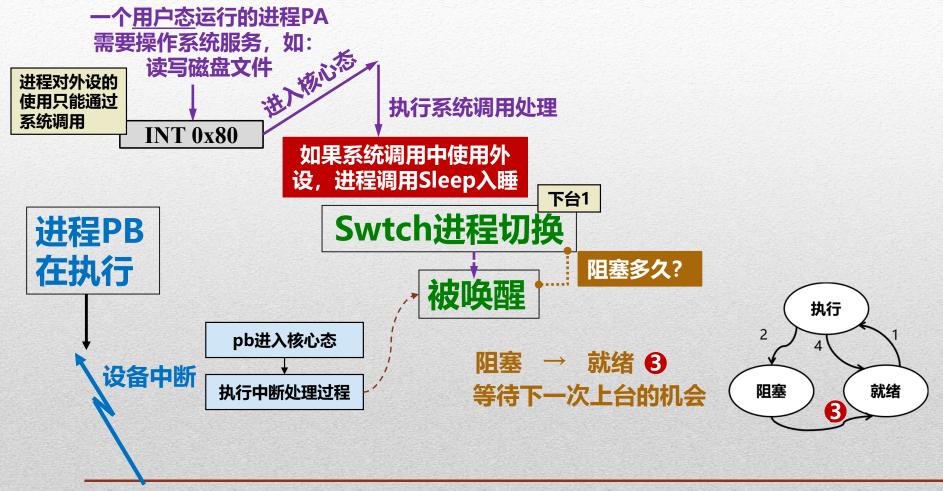


Tongji University, 2022 Fang Yu

19



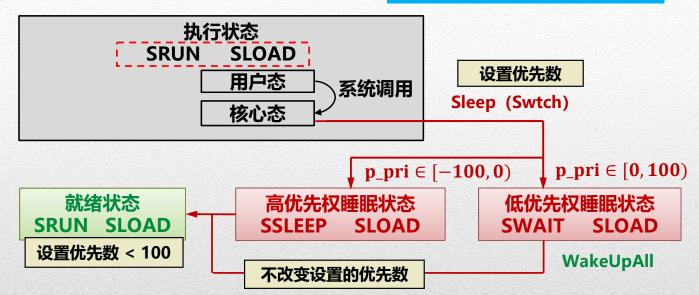
# 重新回看中断与系统调用的流程



# UNIX进程的调度状态

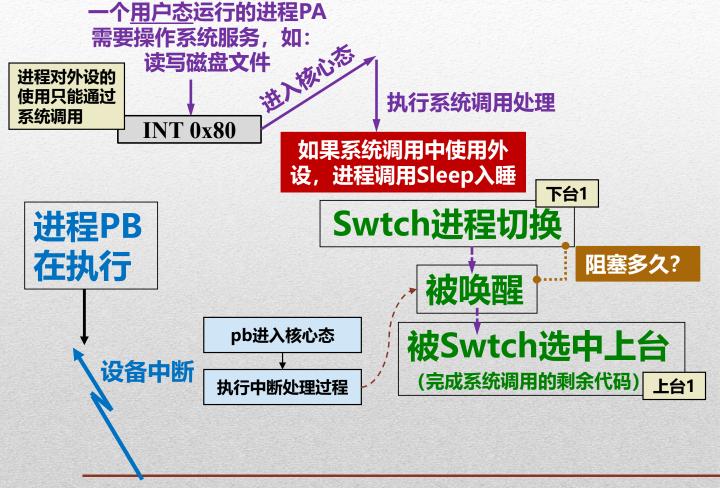


p\_stat=SRUN p\_wchan = 0





# 重新回看中断与系统调用的流程





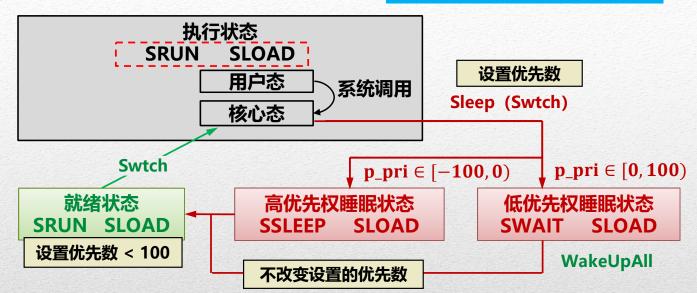
# 重新回看中断与系统调用的流程

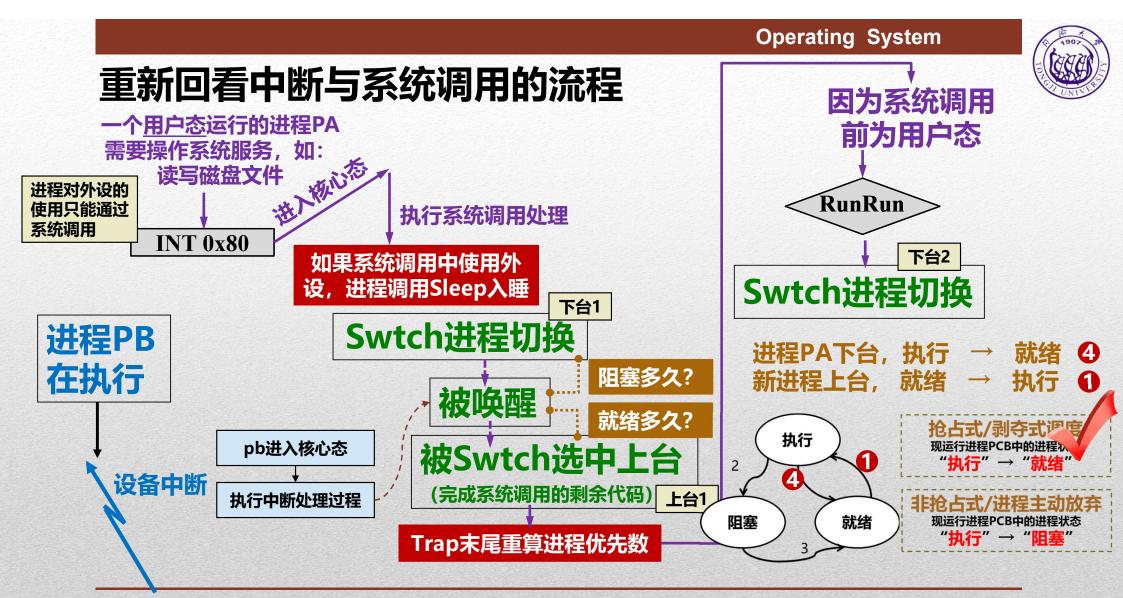


# UNIX进程的调度状态



p\_stat=SRUN p\_wchan = 0

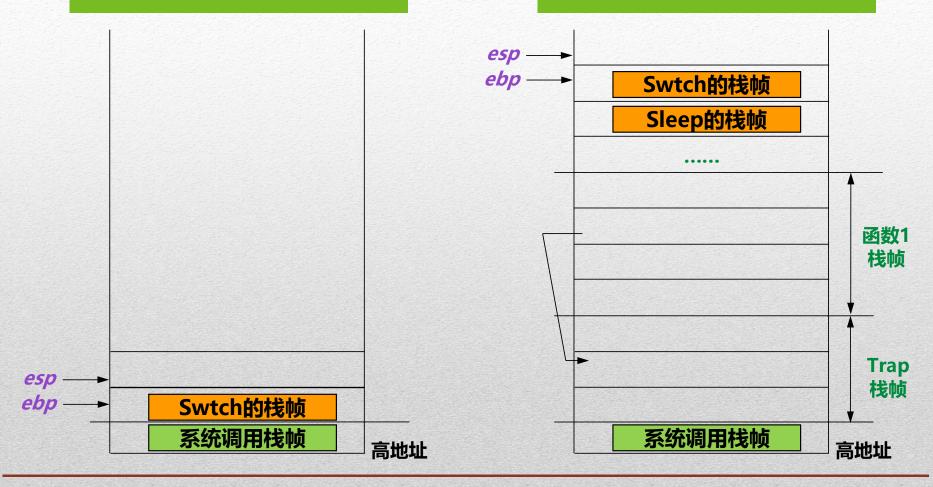




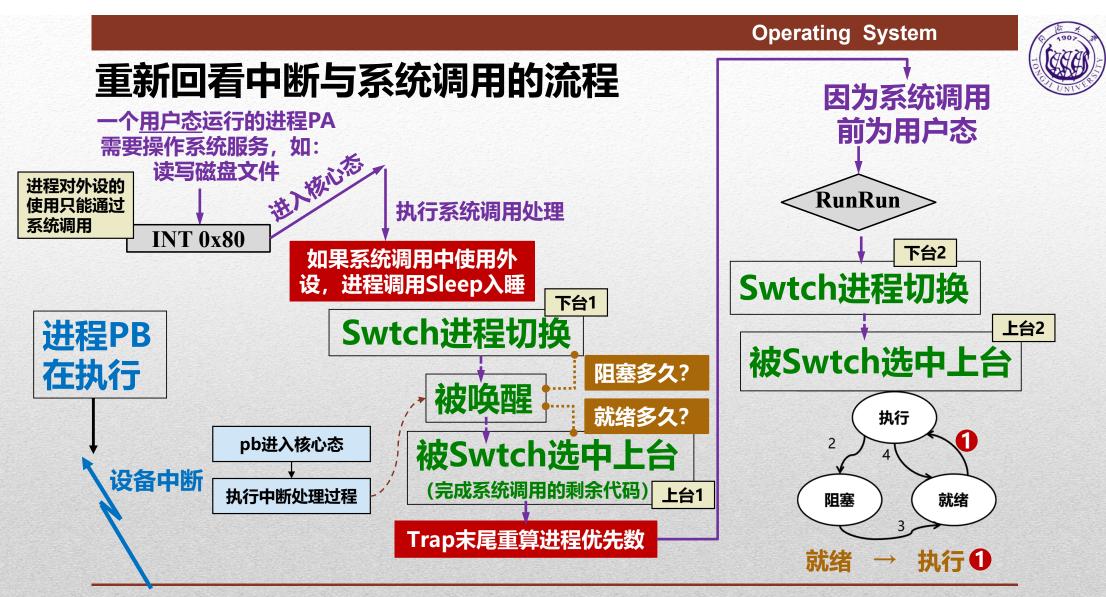


#### 进程抢占调度下台的核心栈

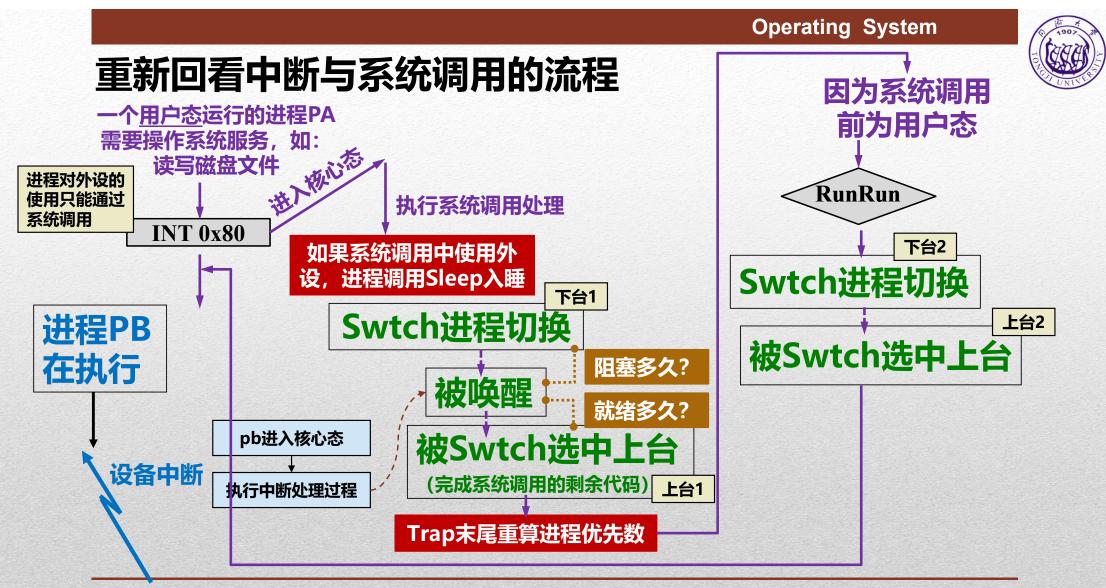
### 进程非抢占调度下台的核心栈

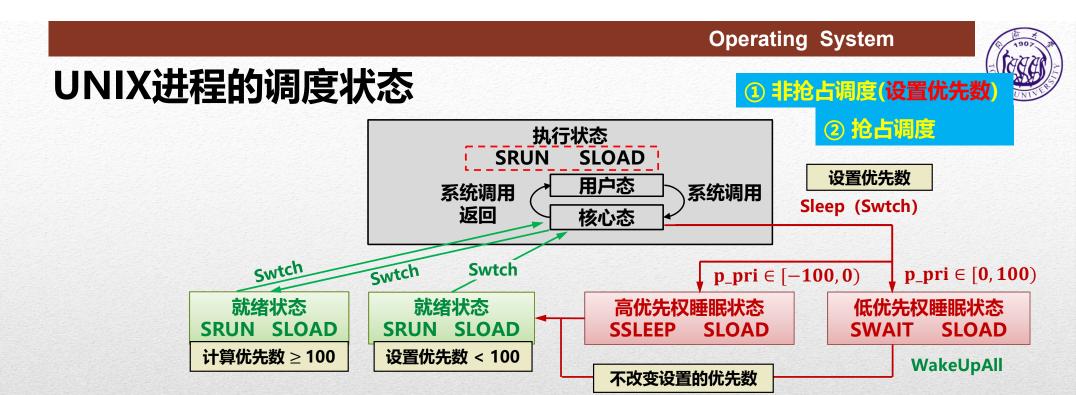


#### **Operating System** UNIX进程的调度状态 ① 非抢占调度(设置优先数) ② 抢占调度 执行状态 SLOAD **SRUN** 设置优先数 用户态 系统调用 Sleep (Swtch) 核心态 Swtch **Swtch** $p_{pri} \in [0, 100)$ $p_{pri} \in [-100, 0)$ 就绪状态 就绪状态 高优先权睡眠状态 低优先权睡眠状态 SRUN SLOAD SRUN SLOAD **SWAIT SSLEEP SLOAD SLOAD** 设置优先数 < 100 计算优先数 ≥ 100 WakeUpAll 不改变设置的优先数



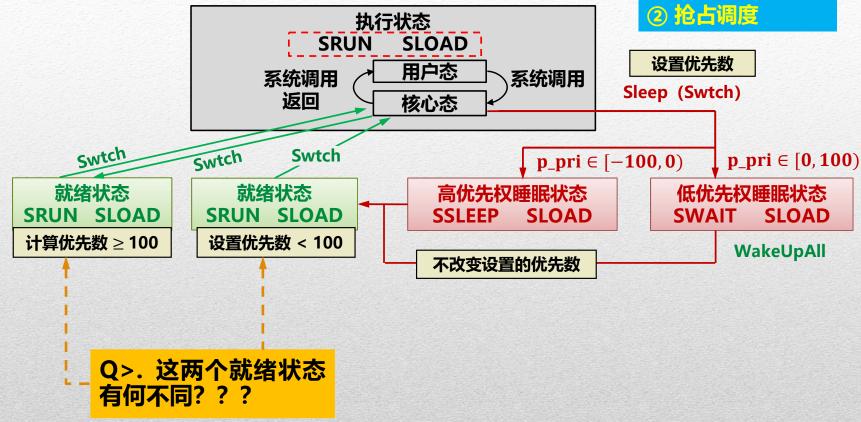
#### **Operating System** UNIX进程的调度状态 ① 非抢占调度(设置优先数) ② 抢占调度 执行状态 SLOAD **SRUN** 设置优先数 用户态 系统调用 Sleep (Swtch) 核心态 Swtch Swtch **Swtch** $p_{pri} \in [0, 100)$ $p_{pri} \in [-100, 0)$ 就绪状态 就绪状态 高优先权睡眠状态 低优先权睡眠状态 SRUN SLOAD SRUN SLOAD **SWAIT SSLEEP SLOAD SLOAD** 设置优先数 < 100 计算优先数 ≥ 100 WakeUpAll 不改变设置的优先数





# UNIX进程的调度状态



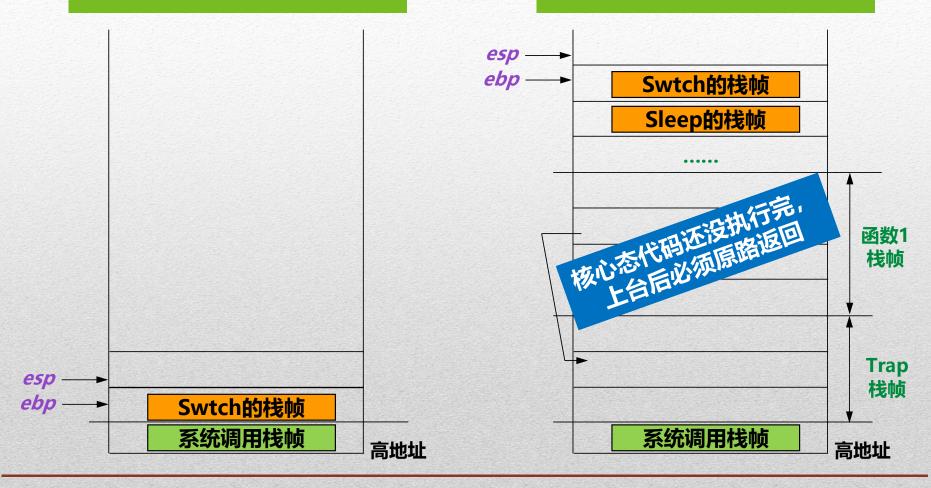


A>. 不同点1: 两个就绪状态的进程的核心栈不同



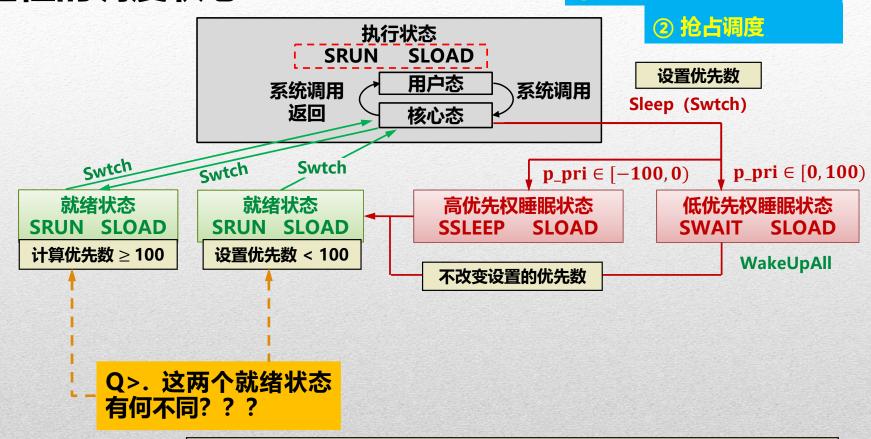
#### 进程抢占调度下台的核心栈

#### 进程非抢占调度下台的核心栈

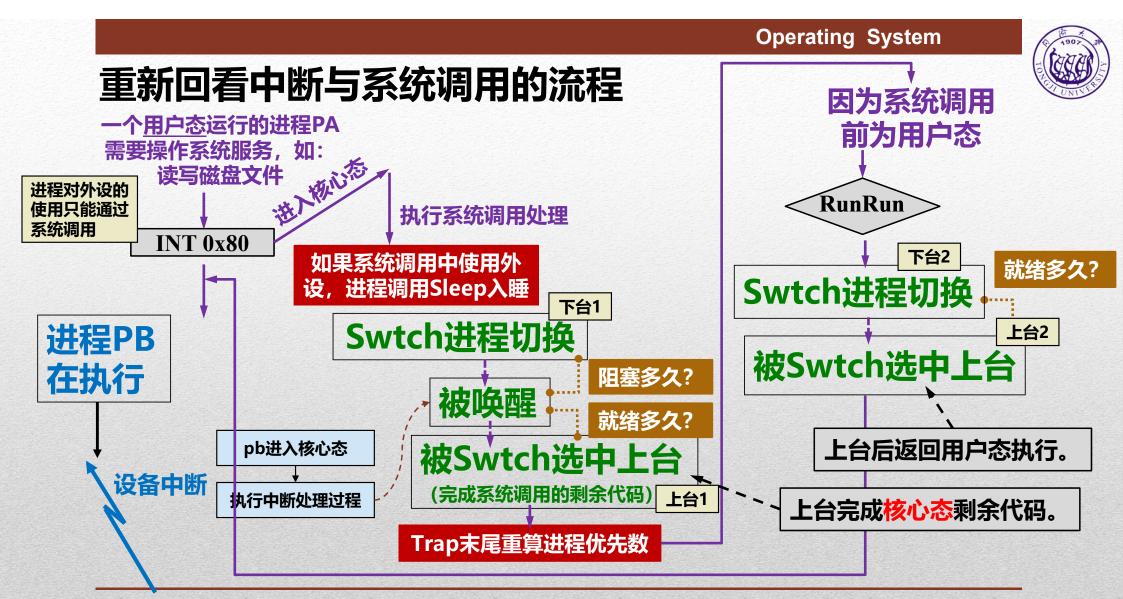


# UNIX进程的调度状态



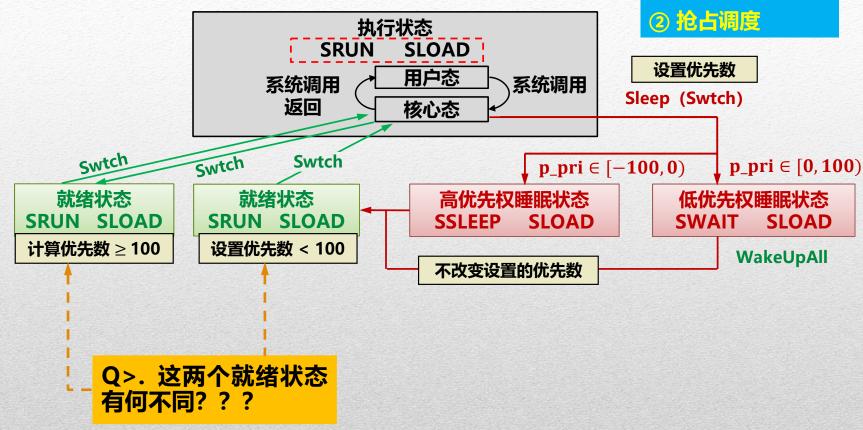


A>. 不同点2: 两个就绪状态的进程再次上台后的任务不同

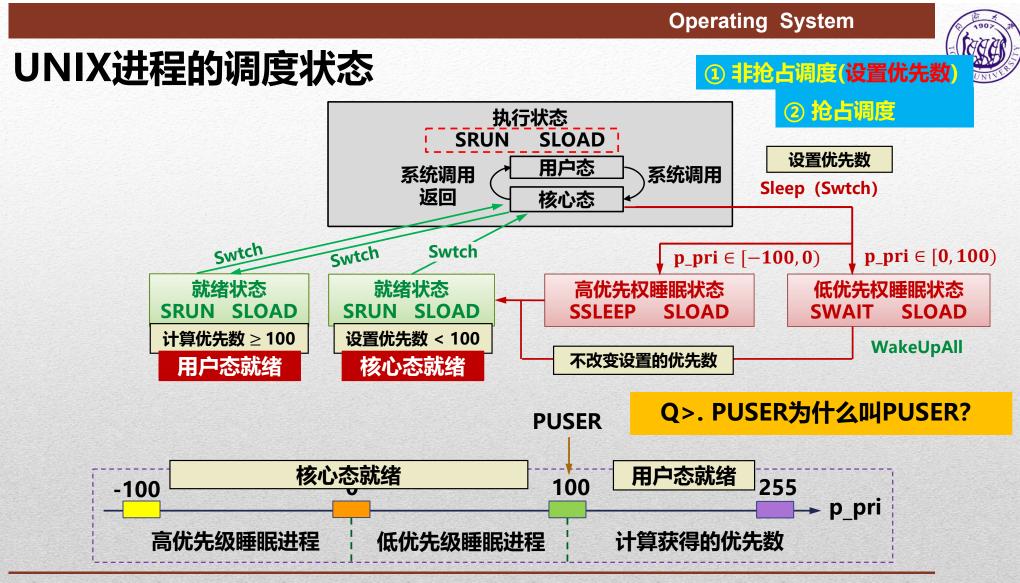


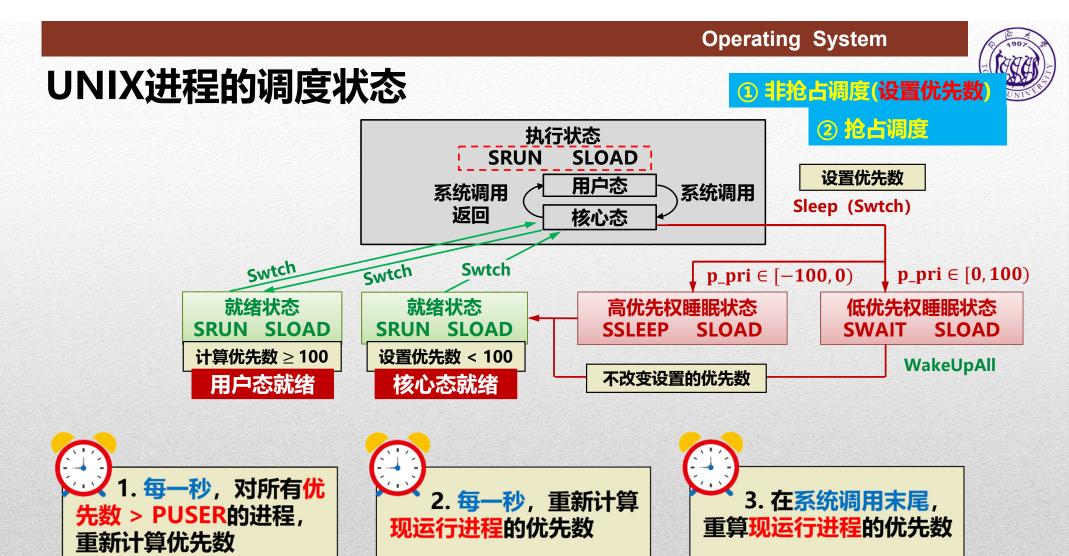
# UNIX进程的调度状态

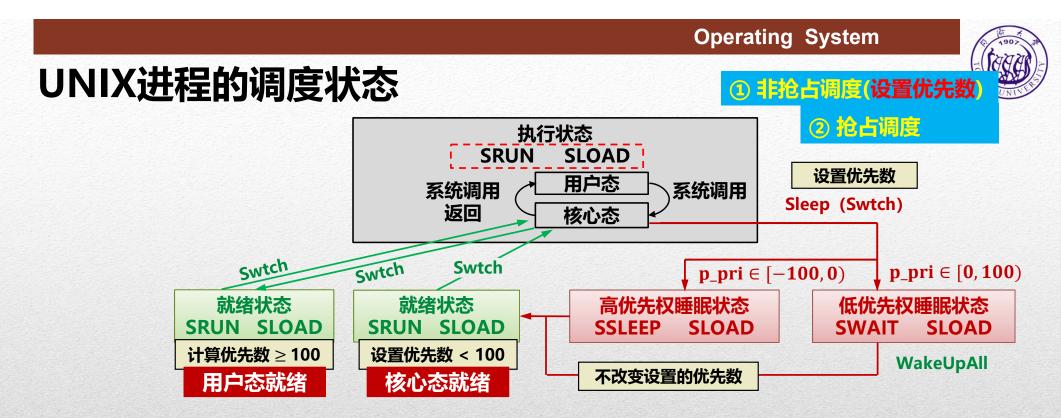




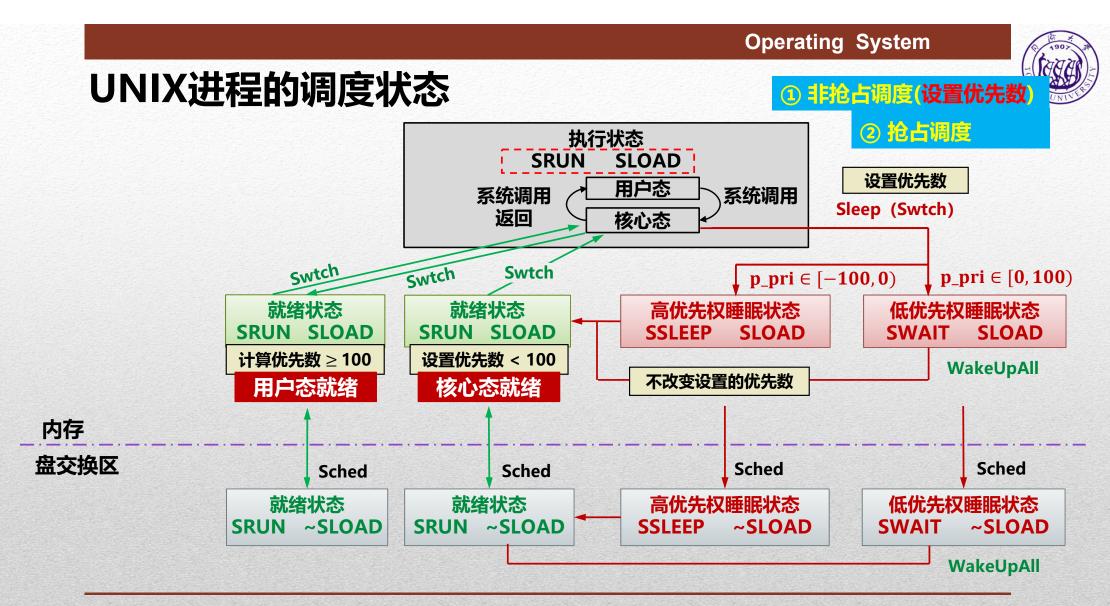
A>. 不同点3: 两个就绪状态的进程优先数不同

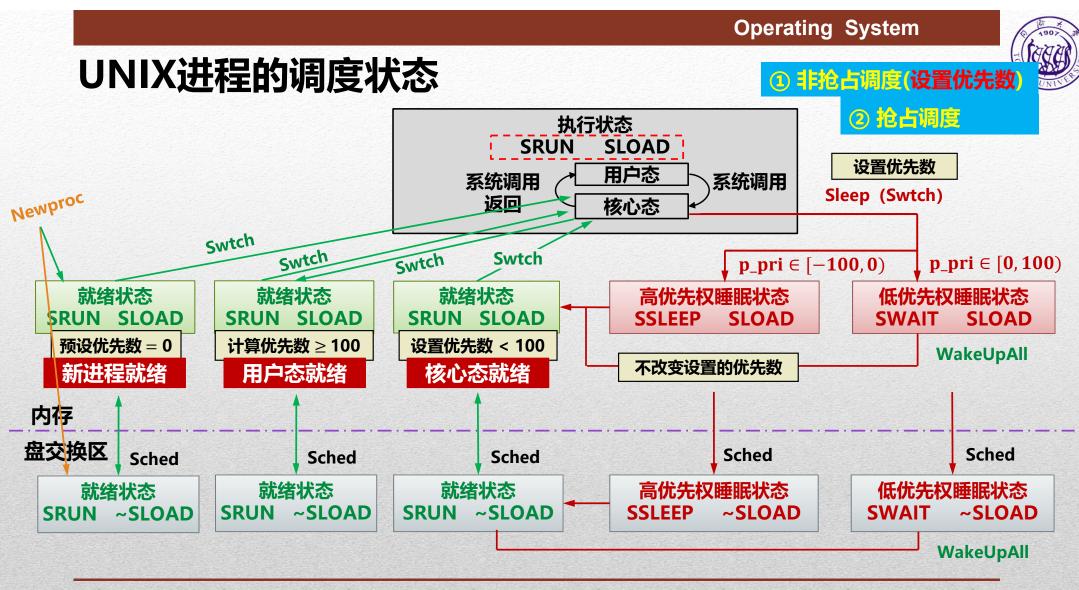




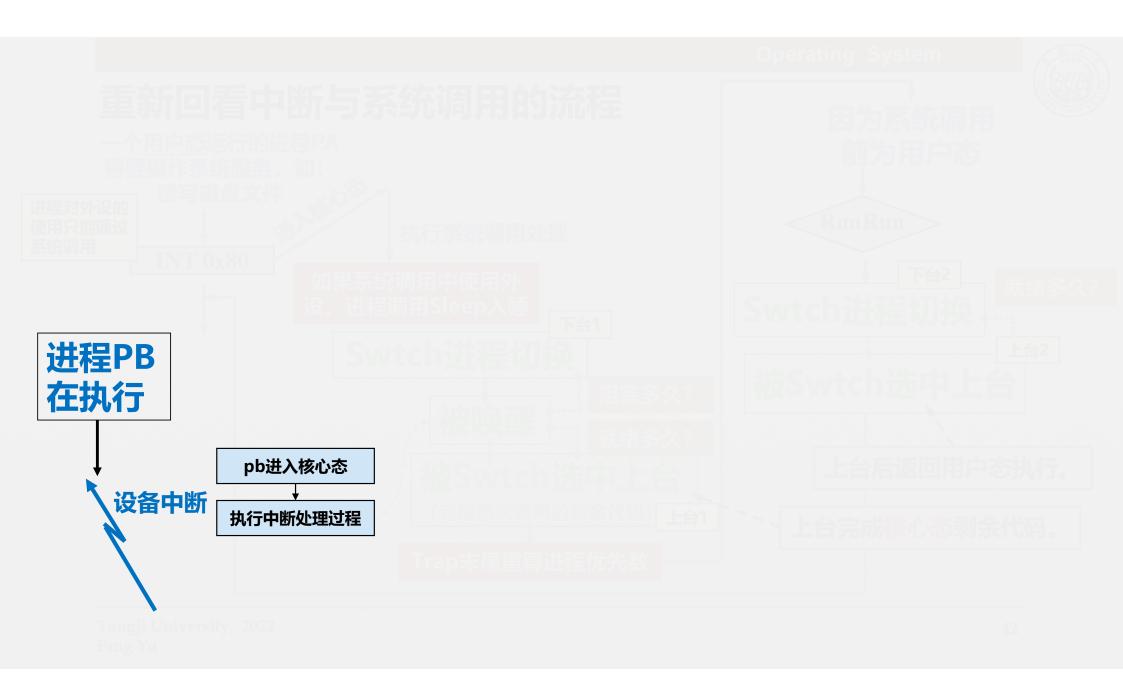


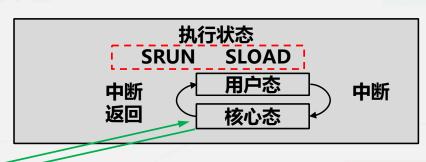
# 如果内存不足以容纳所有的进程图象???





41





Swtch Swtch

SRUN SLOAD 计算优先数 ≥ 100

用户态就绪

**Sched** 

就绪状态 SRUN ~SLOAD

