

汇编语言程序设计

80X86计算机组织结构简述

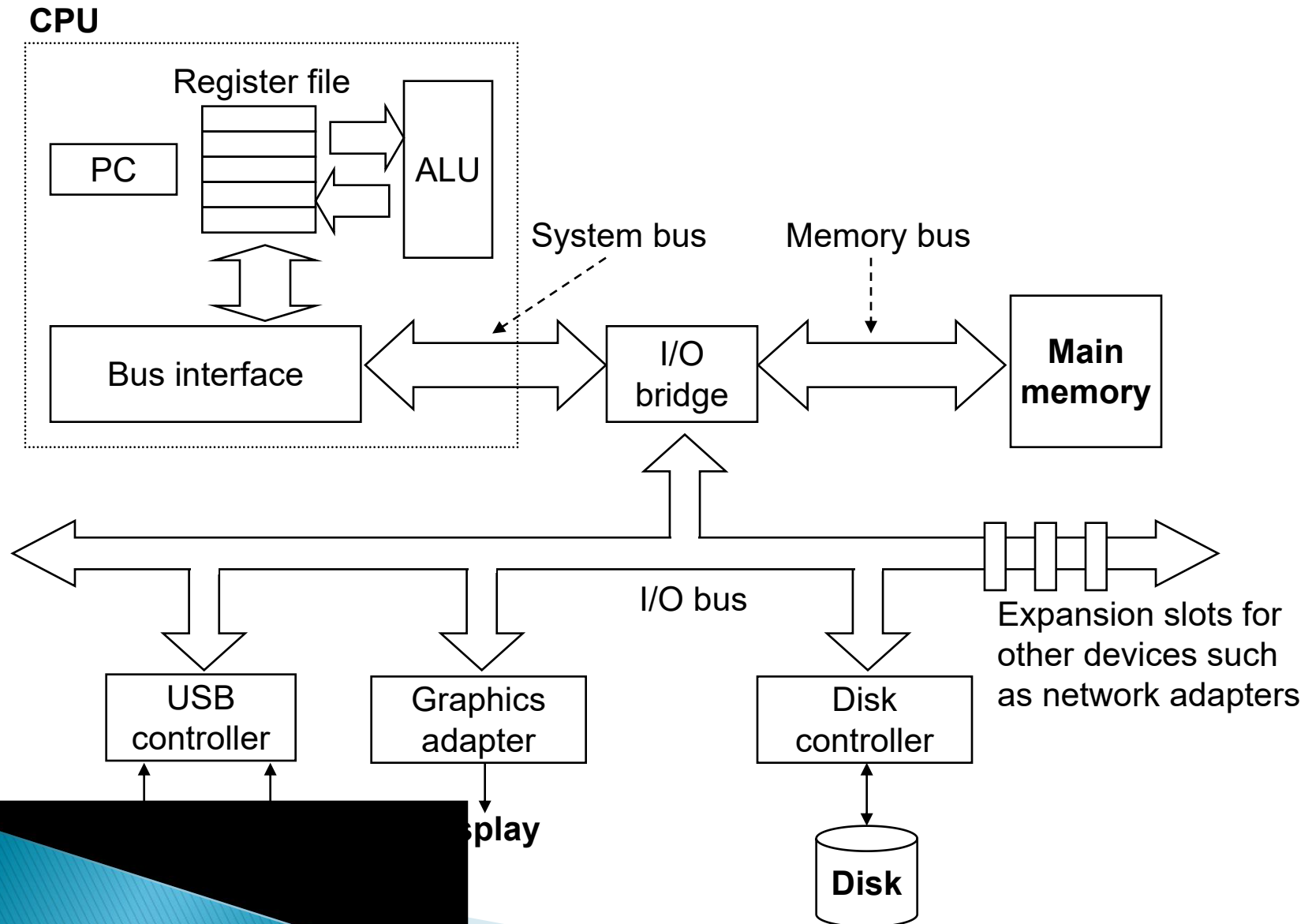


主要内容：

- 计算机系统
- 存储器
- 80X86处理器与保护模式

1. 计算机系统

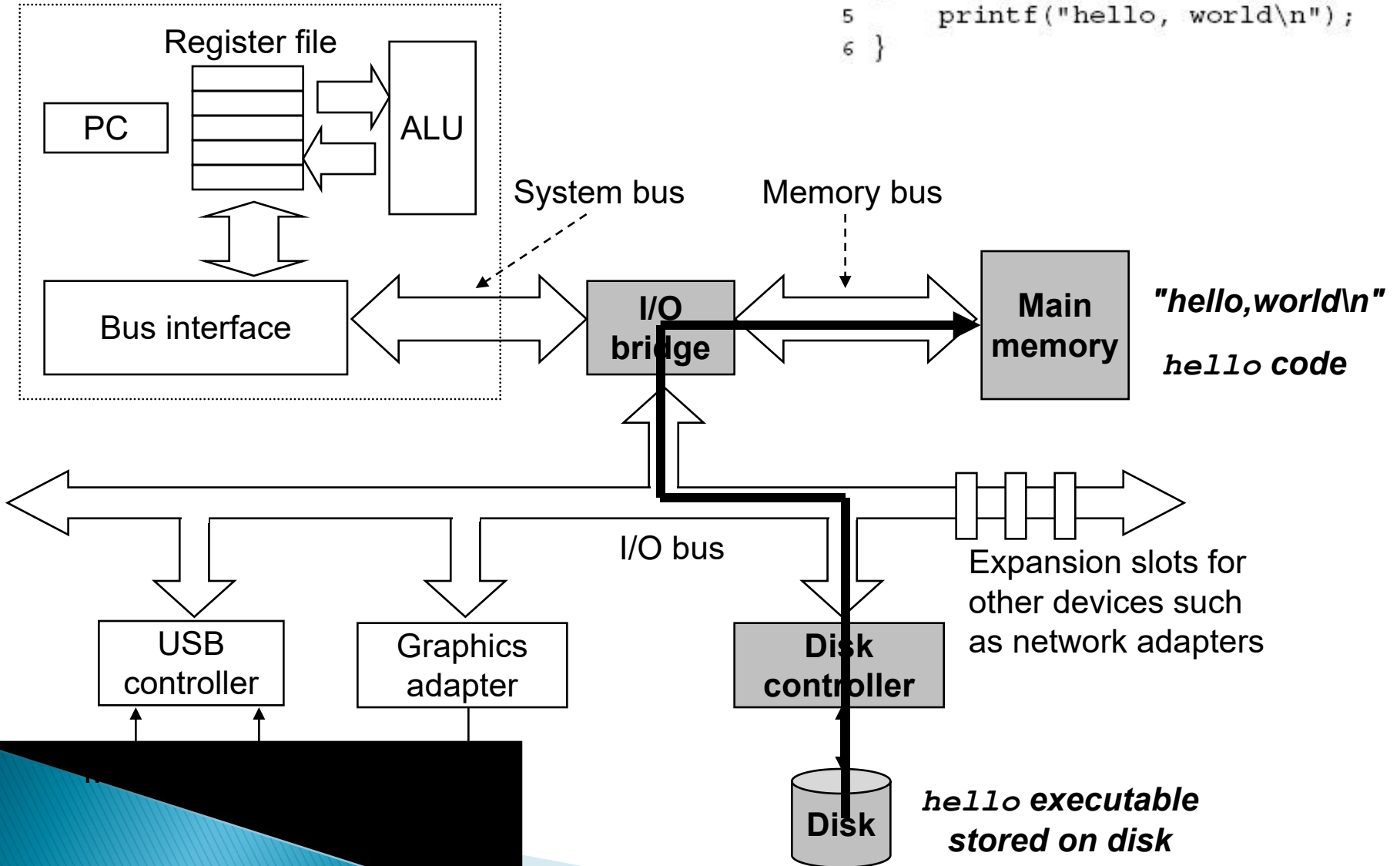
hardware



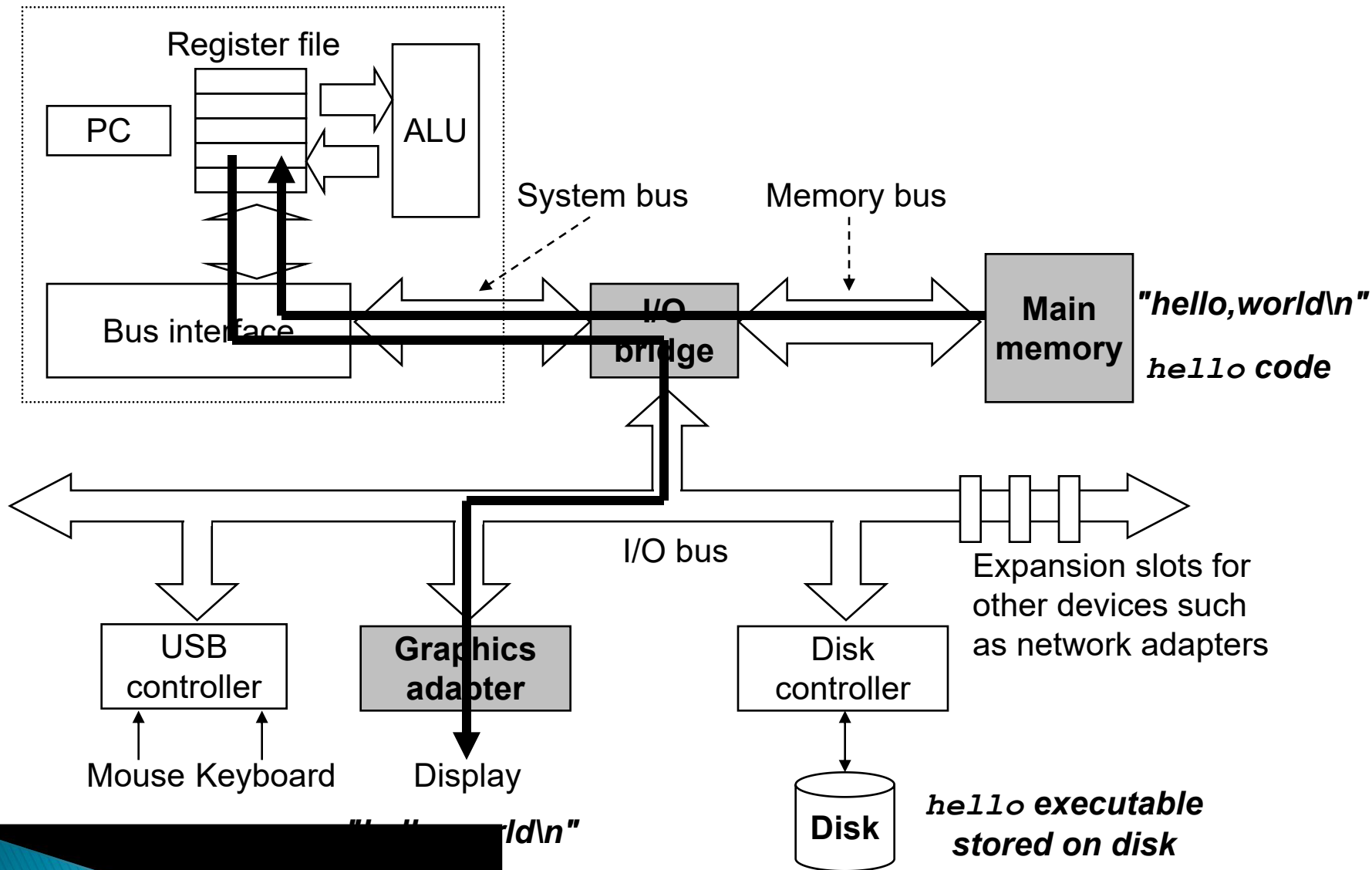
hello.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }
```

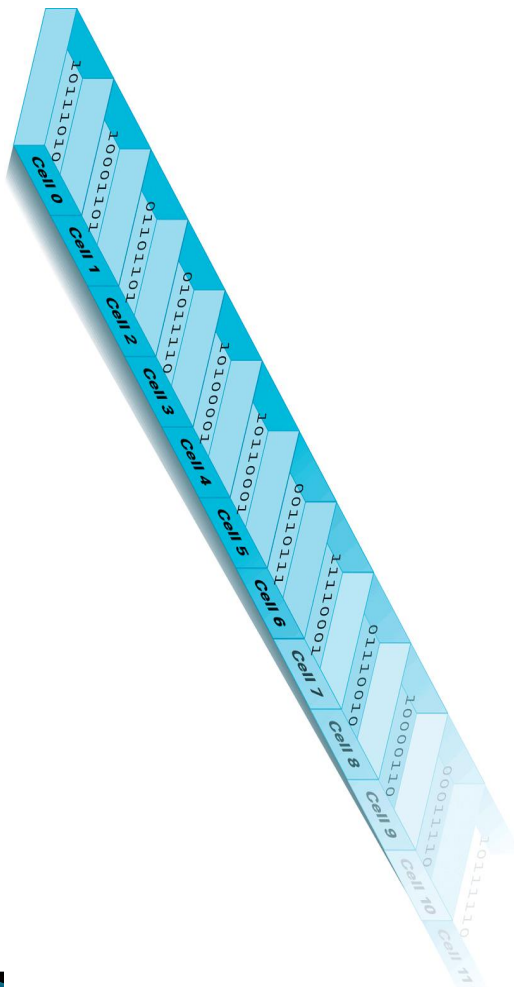
CPU



CPU



2. 主存 (main memory)



存储单元的地址和内容：

- ✓ 存储器以字节 (8 bit) 为单位存储信息
- ✓ 每个字节单元有一个地址，从 0 编号，顺序加 1
- ✓ 地址用二进制数表示 (无符号整数，写成十六进制)
- ✓ 一个32位字要占用相继的四个字节
 - ✓ 低位字节存入低地址，高位字节存入高地址
- ✓ 机器以字对齐地址访问 (读 / 写) 存储器
字单元地址用它的低地址来表示

3. 80X86处理器与保护模式

16位80X86微处理器

1. 8086 / 8088 微处理器

- } 8086是由Intel于1978年设计的微处理器
- } Intel公司在推出8086之后， 推出了介于16位与8位之间的准16位微处理器8088。
- } 8088与8086之间的区别主要在于8088对外只有8根数据线引脚， 访问16位的操作数需要二个总线周期。
- } 8088的这一特点使它能够十分方便地与8位接口芯片相连接。
- } 1980年， IBM公司使用8088成功地开发了16位微型计算机—— IBM-PC。

2. 80186和80286微处理器

- } Intel公司把大型计算机的技术融合到微处理器中, 首先研制的80186在技术上并不十分成熟, 没有获得广泛的应用。
- } 1982年Intel推出了增强型16微处理器 80286, 集成度达13万管 / 片, 时钟频率提高到5MHz~25MHz, 它的16条数据线和24条地址线相互独立, 不再分时使用, 可以寻址16M的地址空间。
- } 80286CPU增加了运行多任务所需要的任务切换、存储管理和多种保护功能。

80286 CPU基本工作方式：

} 实地址方式：

和8086一样，使用20根地址线寻址1M的内存空间，DOS应用程序占用全部系统资源。

} 保护方式：

80286CPU具有虚拟内存管理和多任务处理功能，通过硬件控制可以在多任务之间进行快速切换。

} 80286CPU的内部组成：

➤ 总线接口部件BIU：

地址单元AU、指令单元IU、总线单元BU。

➤ 执行部件EU：

} IBM公司以80286为CPU生产了著名的IBM-PC/AT微型计算机，它的许多技术被沿用至今。

32位80X86微处理器

1. 80386微处理器

- } 1985年，Intel公司推出了第四代微处理器，32位的微处理器80386。
- } 片内集成27.5万个晶体管，时钟频率为16MHz~33 MHz。具有32位数据线和32位地址线，32位通用寄存器。
- } 80386内部由中央处理器CPU、存储器管理部件MMU、总线接口部件BIU组成。
- } 80386有3种工作模式：
实地址模式、虚地址保护模式和虚拟8086模式。

80x86的三种工作模式

1. 实模式

操作相当于一个可进行32位快速运算的8086。

2. 保护模式

是80x86设计目标全部达到的工作模式，通过对程序使用的存储区采用分段、分页的存储管理机制，达到分级使用、互不干扰的保护目的。能为每个任务提供一台虚拟处理器，使每个任务单独执行，快速切换。

3. 虚拟8086模式

保护模式下同时模拟多个8086处理器。

32位微处理器的寄存器

- 80X86微处理器由16位升级为32位后，它的寄存器也对应升级为32位。
- 为了新的工作方式和存储管理的需要，增加了一些用于控制的寄存器。

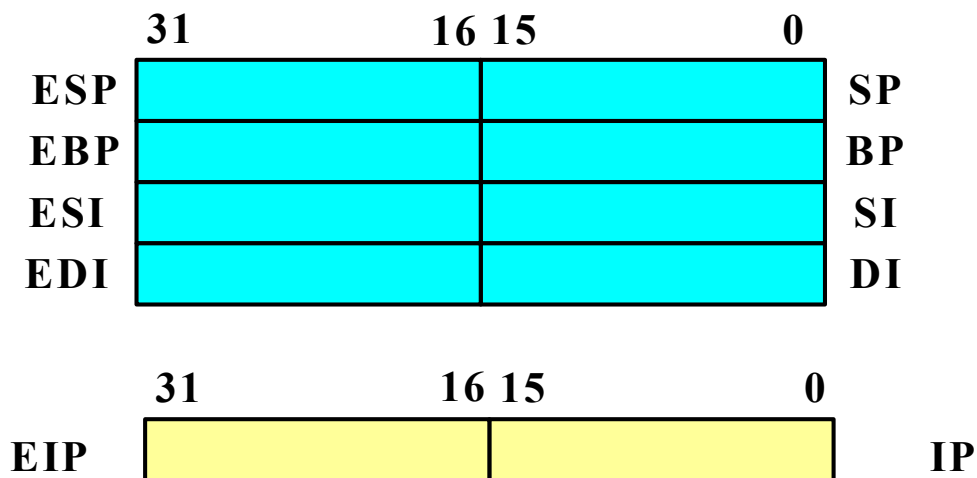
1. 数据寄存器

16位80X86处理器原有的4个通用数据寄存器扩展为32位，命名为EAX、EBX、ECX和EDX。仍然可以使用原有的16位和8位寄存器，如AX、BX、CX、DX、AH、AL、BH、BL.....。

	31	16	15	8	7	0	
EAX					AH	AL	AX
EBX					BH	BL	BX
					CH	CL	CX
					DH	DL	DX

2. 地址寄存器

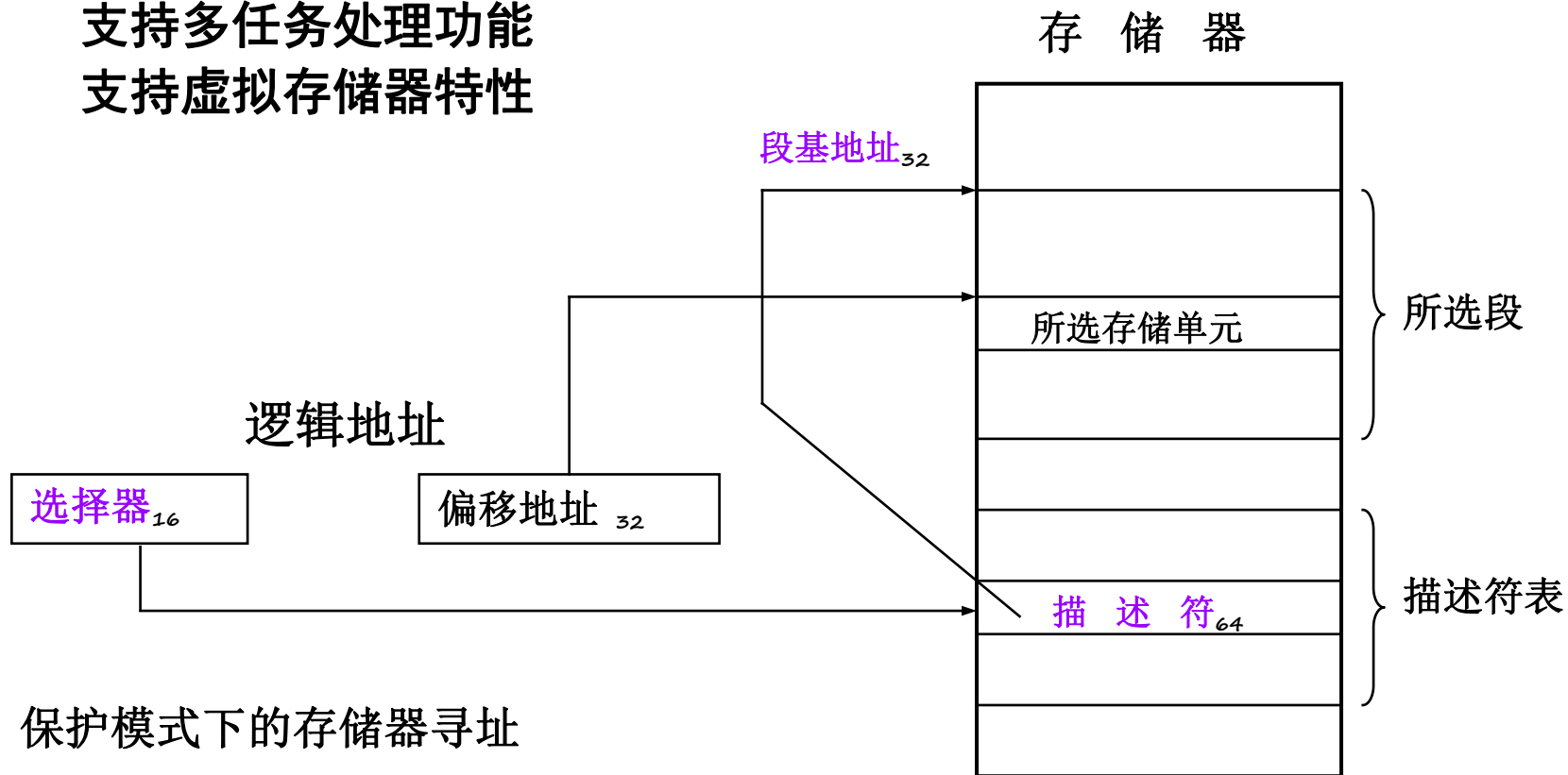
- 原有的4个主要用于内存寻址的通用寄存器同样扩展为32位，命名为ESI、EDI、EBP、ESP。在实地址模式下仍然可以使用原有的16位寄存器SI、DI、BP和SP。
- 指令指针寄存器扩展为32位，更名为EIP，实地址下仍然可以使用它的低16位IP。



- } 在原有的4个段寄存器（CS DS SS ES）基础上，增加了2个新的段寄存器FS和GS。
- } 段寄存器长度均为16位，其中13位代表内存段的一个编号，称为“段选择器”。

保护模式下的80x86（段模式）

支持多任务处理功能
支持虚拟存储器特性



保护模式保护什么？

分清不同程序使用的存储区域，不允许随便使用别人的数据和代码。

必要条件：

1. 要标记每段存储区的所有者或被使用的权限级别。
2. 要标记使用者是谁（权限级别）。
3. 中间环节：CPU要去判断此次访问是否合法。

- } 在X86-32体系结构的保护模式下，一个内存地址是由段基地址、偏移地址两个要素构成的。
- } 每个段的描述（即段描述符）由三个要素构成——段基地址（32位）：段长度（20位，段长度单位为 2^{12} ）：访问权限。
 - } 段描述符的长度为64位
- } 出于系统兼容原因，段寄存器只有16位，如何表示64位的段描述符？
 - } 通过描述符表：将段寄存器中的高13 位值作为索引来访问该表，从而获得64位的段描述符。

} GDT是全局描述符表，主要存放操作系统和各任务公用的描述符

- 公用的数据和代码段描述符、各任务的TSS描述符和LDT描述符

✎ TSS是任务状态段，存放各个任务私有运行状态信息描述符

- GDT register (GDTR) , 48bit

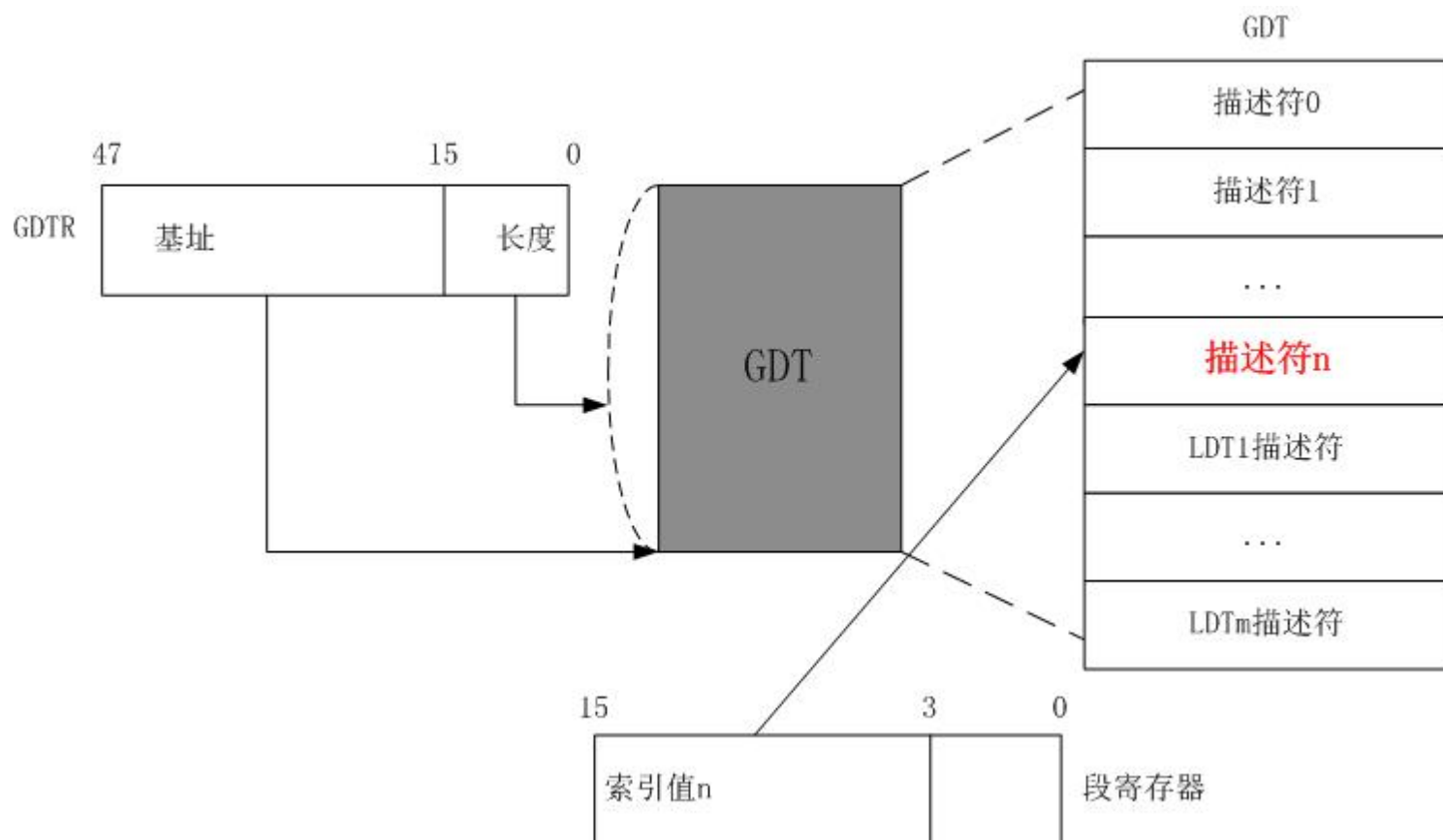
} LDT是局部描述符表，主要存放各个任务的私有描述符

} 段寄存器：高13位用来指示描述符在描述符表中的索引号，低两位是表示使用描述符的特权级别。

- 另外一位（T1）是GDT和LDT的信号量，如果T1=0，则使用GDT，如果T1=1，则使用LDT

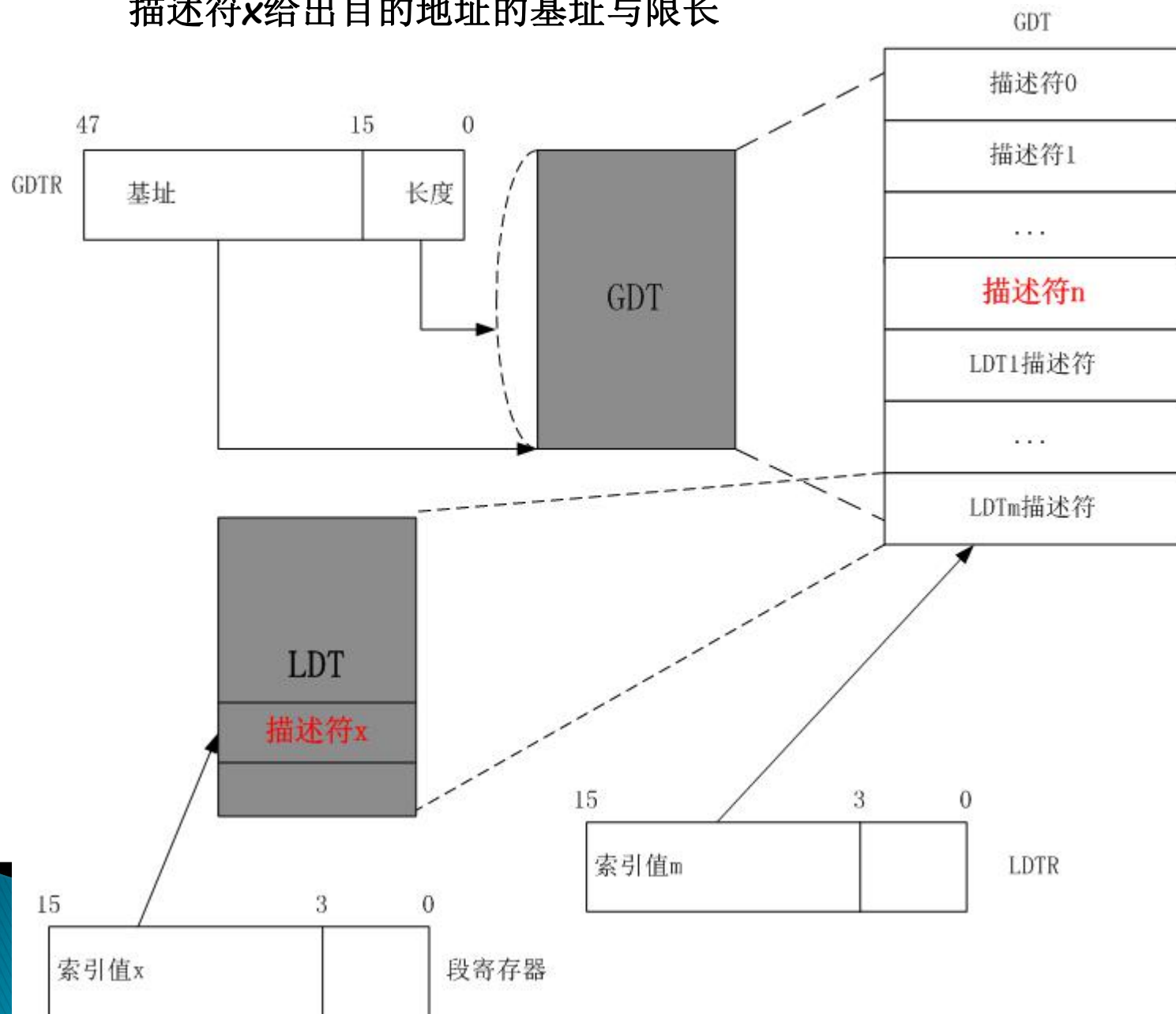
$T1 = 0$

描述符 n 给出目的地址的基址与限长



$T1 = 1$

描述符x给出目的地址的基址与限长



寄存器与存储器的比较：

寄 存 器	存 储 器
在CPU内部 访问速度快 容量小，成本高 用名字表示 没有地址	在CPU外部 访问速度慢 容量大，成本低 用地址表示 地址可用各种方式形成

