

目录

C O N T E N T S

出版者的话
中文版序一
中文版序二
译者序
前言
关于作者

第 1 章 计算机系统漫游	1	1.7.1 进程	11
1.1 信息就是位+上下文	1	1.7.2 线程	12
1.2 程序被其他程序翻译成不同的 格式	3	1.7.3 虚拟内存	12
1.3 了解编译系统如何工作是 大有益处的	4	1.7.4 文件	14
1.4 处理器读并解释储存在内存 中的指令	5	1.8 系统之间利用网络通信	14
1.4.1 系统的硬件组成	5	1.9 重要主题	16
1.4.2 运行 hello 程序	7	1.9.1 Amdahl 定律	16
1.5 高速缓存至关重要	9	1.9.2 并发和并行	17
1.6 存储设备形成层次结构	9	1.9.3 计算机系统中抽象的 重要性	19
1.7 操作系统管理硬件	10	1.10 小结	20
		参考文献说明	20
		练习题答案	20

第一部分 程序结构和执行

第 2 章 信息的表示和处理	22	2.1.3 寻址和字节顺序	29
2.1 信息存储	24	2.1.4 表示字符串	34
2.1.1 十六进制表示法	25	2.1.5 表示代码	34
2.1.2 字数据大小	27	2.1.6 布尔代数简介	35

2.1.7 C语言中的位级运算	37	3.2.3 关于格式的注解	117
2.1.8 C语言中的逻辑运算	39	3.3 数据格式	119
2.1.9 C语言中的移位运算	40	3.4 访问信息	119
2.2 整数表示	41	3.4.1 操作数指示符	121
2.2.1 整型数据类型	42	3.4.2 数据传送指令	122
2.2.2 无符号数的编码	43	3.4.3 数据传送示例	125
2.2.3 补码编码	44	3.4.4 压入和弹出栈数据	127
2.2.4 有符号数和无符号数之间的 转换	49	3.5 算术和逻辑操作	128
2.2.5 C语言中的有符号数与 无符号数	52	3.5.1 加载有效地址	129
2.2.6 扩展一个数字的位表示	54	3.5.2 一元和二元操作	130
2.2.7 截断数字	56	3.5.3 移位操作	131
2.2.8 关于有符号数与无符号数的 建议	58	3.5.4 讨论	131
2.3 整数运算	60	3.5.5 特殊的算术操作	133
2.3.1 无符号加法	60	3.6 控制	135
2.3.2 补码加法	62	3.6.1 条件码	135
2.3.3 补码的非	66	3.6.2 访问条件码	136
2.3.4 无符号乘法	67	3.6.3 跳转指令	138
2.3.5 补码乘法	67	3.6.4 跳转指令的编码	139
2.3.6 乘以常数	70	3.6.5 用条件控制来实现条件分支	141
2.3.7 除以2的幂	71	3.6.6 用条件传送来实现条件分支	145
2.3.8 关于整数运算的最后思考	74	3.6.7 循环	149
2.4 浮点数	75	3.6.8 switch 语句	159
2.4.1 二进制小数	76	3.7 过程	164
2.4.2 IEEE 浮点表示	78	3.7.1 运行时栈	164
2.4.3 数字示例	79	3.7.2 转移控制	165
2.4.4 舍入	83	3.7.3 数据传送	168
2.4.5 浮点运算	85	3.7.4 栈上的局部存储	170
2.4.6 C语言中的浮点数	86	3.7.5 寄存器中的局部存储空间	172
2.5 小结	87	3.7.6 递归过程	174
参考文献说明	88	3.8 数组分配和访问	176
家庭作业	88	3.8.1 基本原则	176
练习题答案	97	3.8.2 指针运算	177
第3章 程序的机器级表示	109	3.8.3 嵌套的数组	178
3.1 历史观点	110	3.8.4 定长数组	179
3.2 程序编码	113	3.8.5 变长数组	181
3.2.1 机器级代码	113	3.9 异质的数据结构	183
3.2.2 代码示例	114	3.9.1 结构	183
		3.9.2 联合	186
		3.9.3 数据对齐	189
		3.10 在机器级程序中将控制与 数据结合起来	192

3.10.1 理解指针	192	4.4 流水线的通用原理	282
3.10.2 应用: 使用 GDB 调试器 ...	193	4.4.1 计算流水线	282
3.10.3 内存越界引用和缓冲区 溢出	194	4.4.2 流水线操作的详细说明 ...	284
3.10.4 对抗缓冲区溢出攻击	198	4.4.3 流水线的局限性	284
3.10.5 支持变长栈帧	201	4.4.4 带反馈的流水线系统	287
3.11 浮点代码	204	4.5 Y86-64 的流水线实现	288
3.11.1 浮点传送和转换操作	205	4.5.1 SEQ+: 重新安排计算 阶段	288
3.11.2 过程中的浮点代码	209	4.5.2 插入流水线寄存器	289
3.11.3 浮点运算操作	210	4.5.3 对信号进行重新排列和 标号	292
3.11.4 定义和使用浮点常数	212	4.5.4 预测下一个 PC	293
3.11.5 在浮点代码中使用位级 操作	212	4.5.5 流水线冒险	295
3.11.6 浮点比较操作	213	4.5.6 异常处理	306
3.11.7 对浮点代码的观察结论 ...	215	4.5.7 PIPE 各阶段的实现	308
3.12 小结	216	4.5.8 流水线控制逻辑	314
参考文献说明	216	4.5.9 性能分析	322
家庭作业	216	4.5.10 未完成的工作	323
练习题答案	226	4.6 小结	325
第 4 章 处理器体系结构	243	参考文献说明	326
4.1 Y86-64 指令集体系结构	245	家庭作业	327
4.1.1 程序员可见的状态	245	练习题答案	331
4.1.2 Y86-64 指令	245	第 5 章 优化程序性能	341
4.1.3 指令编码	246	5.1 优化编译器的能力和局限性 ...	342
4.1.4 Y86-64 异常	250	5.2 表示程序性能	345
4.1.5 Y86-64 程序	251	5.3 程序示例	347
4.1.6 一些 Y86-64 指令的详情	255	5.4 消除循环的低效率	350
4.2 逻辑设计和硬件控制语言 HCL ...	256	5.5 减少过程调用	353
4.2.1 逻辑门	257	5.6 消除不必要的内存引用	354
4.2.2 组合电路和 HCL 布尔 表达式	257	5.7 理解现代处理器	357
4.2.3 字级的组合电路和 HCL 整数表达式	258	5.7.1 整体操作	357
4.2.4 集合关系	261	5.7.2 功能单元的性能	361
4.2.5 存储器和时钟	262	5.7.3 处理器操作的抽象模型 ...	362
4.3 Y86-64 的顺序实现	264	5.8 循环展开	366
4.3.1 将处理组织成阶段	264	5.9 提高并行性	369
4.3.2 SEQ 硬件结构	272	5.9.1 多个累积变量	370
4.3.3 SEQ 的时序	274	5.9.2 重新结合变换	373
4.3.4 SEQ 阶段的实现	277	5.10 优化合并代码的结果小结	377
		5.11 一些限制因素	378
		5.11.1 寄存器溢出	378

5.11.2 分支预测和预测错误 处罚	379	6.3 存储器层次结构	421
5.12 理解内存性能	382	6.3.1 存储器层次结构中的缓存	422
5.12.1 加载的性能	382	6.3.2 存储器层次结构概念小结	424
5.12.2 存储的性能	383	6.4 高速缓存存储器	425
5.13 应用：性能提高技术	387	6.4.1 通用的高速缓存存储器 组织结构	425
5.14 确认和消除性能瓶颈	388	6.4.2 直接映射高速缓存	427
5.14.1 程序剖析	388	6.4.3 组相联高速缓存	433
5.14.2 使用剖析程序来指导 优化	390	6.4.4 全相联高速缓存	434
5.15 小结	392	6.4.5 有关写的问题	437
参考文献说明	393	6.4.6 一个真实的高速缓存层次 结构的解剖	438
家庭作业	393	6.4.7 高速缓存参数的性能影响	439
练习题答案	395	6.5 编写高速缓存友好的代码	440
第6章 存储器层次结构	399	6.6 综合：高速缓存对程序性能的 影响	444
6.1 存储技术	399	6.6.1 存储器山	444
6.1.1 随机访问存储器	400	6.6.2 重新排列循环以提高空间 局部性	447
6.1.2 磁盘存储	406	6.6.3 在程序中利用局部性	450
6.1.3 固态硬盘	414	6.7 小结	450
6.1.4 存储技术趋势	415	参考文献说明	451
6.2 局部性	418	家庭作业	451
6.2.1 对程序数据引用的局部性	418	练习题答案	459
6.2.2 取指令的局部性	419		
6.2.3 局部性小结	420		

第二部分

在系统上运行程序

第7章 链接	464	7.7 重定位	478
7.1 编译器驱动程序	465	7.7.1 重定位条目	479
7.2 静态链接	466	7.7.2 重定位符号引用	479
7.3 目标文件	466	7.8 可执行目标文件	483
7.4 可重定位目标文件	467	7.9 加载可执行目标文件	484
7.5 符号和符号表	468	7.10 动态链接共享库	485
7.6 符号解析	470	7.11 从应用程序中加载和链接 共享库	487
7.6.1 链接器如何解析多重定义 的全局符号	471	7.12 位置无关代码	489
7.6.2 与静态库链接	475	7.13 库打桩机制	492
7.6.3 链接器如何使用静态库来 解析引用	477	7.13.1 编译时打桩	492
		7.13.2 链接时打桩	492

7.13.3 运行时打桩	494
7.14 处理目标文件的工具	496
7.15 小结	496
参考文献说明	497
家庭作业	497
练习题答案	499
第8章 异常控制流	501
8.1 异常	502
8.1.1 异常处理	503
8.1.2 异常的类别	504
8.1.3 Linux/x86-64 系统中的 异常	505
8.2 进程	508
8.2.1 逻辑控制流	508
8.2.2 并发流	509
8.2.3 私有地址空间	509
8.2.4 用户模式和内核模式	510
8.2.5 上下文切换	511
8.3 系统调用错误处理	512
8.4 进程控制	513
8.4.1 获取进程 ID	513
8.4.2 创建和终止进程	513
8.4.3 回收子进程	516
8.4.4 让进程休眠	521
8.4.5 加载并运行程序	521
8.4.6 利用 fork 和 execve 运行 程序	524
8.5 信号	526
8.5.1 信号术语	527
8.5.2 发送信号	528
8.5.3 接收信号	531
8.5.4 阻塞和解除阻塞信号	532
8.5.5 编写信号处理程序	533
8.5.6 同步流以避免讨厌的并发 错误	540
8.5.7 显式地等待信号	543
8.6 非本地跳转	546
8.7 操作进程的工具	550
8.8 小结	550
参考文献说明	550
家庭作业	550

练习题答案	556
-------------	-----

第9章 虚拟内存 559

9.1 物理和虚拟寻址	560
9.2 地址空间	560
9.3 虚拟内存作为缓存的工具	561
9.3.1 DRAM 缓存的组织结构	562
9.3.2 页表	562
9.3.3 页命中	563
9.3.4 缺页	564
9.3.5 分配页面	565
9.3.6 又是局部性救了我们	565
9.4 虚拟内存作为内存管理的 工具	565
9.5 虚拟内存作为内存保护的 工具	567
9.6 地址翻译	567
9.6.1 结合高速缓存和虚拟 内存	570
9.6.2 利用 TLB 加速地址翻译	570
9.6.3 多级页表	571
9.6.4 综合: 端到端的地址翻译	573
9.7 案例研究: Intel Core i7/Linux 内存系统	576
9.7.1 Core i7 地址翻译	576
9.7.2 Linux 虚拟内存系统	580
9.8 内存映射	582
9.8.1 再看共享对象	583
9.8.2 再看 fork 函数	584
9.8.3 再看 execve 函数	584
9.8.4 使用 mmap 函数的用户级 内存映射	585
9.9 动态内存分配	587
9.9.1 malloc 和 free 函数	587
9.9.2 为什么要使用动态内存 分配	589
9.9.3 分配器的要求和目标	590
9.9.4 碎片	591
9.9.5 实现问题	592
9.9.6 隐式空闲链表	592
9.9.7 放置已分配的块	593
9.9.8 分割空闲块	594
9.9.9 获取额外的堆内存	594

9.9.10	合并空闲块	594	9.11.2	读未初始化的内存	609
9.9.11	带边界标记的合并	595	9.11.3	允许栈缓冲区溢出	610
9.9.12	综合：实现一个简单的 分配器	597	9.11.4	假设指针和它们指向的 对象是相同大小的	610
9.9.13	显式空闲链表	603	9.11.5	造成错位错误	611
9.9.14	分离的空闲链表	604	9.11.6	引用指针，而不是它所 指向的对象	611
9.10	垃圾收集	605	9.11.7	误解指针运算	611
9.10.1	垃圾收集器的基本知识 ...	606	9.11.8	引用不存在的变量	612
9.10.2	Mark&Sweep 垃圾 收集器	607	9.11.9	引用空闲堆块中的数据 ...	612
9.10.3	C程序的保守 Mark& Sweep	608	9.11.10	引起内存泄漏	613
9.11	C程序中常见的与内存有关的 错误	609	9.12	小结	613
9.11.1	间接引用坏指针	609	参考文献说明		613
			家庭作业		614
			练习题答案		617

第三部分
程序间的交互和通信

第 10 章	系统级 I/O	622	第 11 章	网络编程	642
10.1	Unix I/O	622	11.1	客户端-服务器编程模型	642
10.2	文件	623	11.2	网络	643
10.3	打开和关闭文件	624	11.3	全球 IP 因特网	646
10.4	读和写文件	625	11.3.1	IP 地址	647
10.5	用 RIO 包健壮地读写	626	11.3.2	因特网域名	649
10.5.1	RIO 的无缓冲的输入输出 函数	627	11.3.3	因特网连接	651
10.5.2	RIO 的带缓冲的输入 函数	627	11.4	套接字接口	652
10.6	读取文件元数据	632	11.4.1	套接字地址结构	653
10.7	读取目录内容	633	11.4.2	socket 函数	654
10.8	共享文件	634	11.4.3	connect 函数	654
10.9	I/O 重定向	637	11.4.4	bind 函数	654
10.10	标准 I/O	638	11.4.5	listen 函数	655
10.11	综合：我该使用哪些 I/O 函数？	638	11.4.6	accept 函数	655
10.12	小结	640	11.4.7	主机和服务的转换	656
参考文献说明		640	11.4.8	套接字接口的辅助函数 ...	660
家庭作业		640	11.4.9	echo 客户端和服务器的 示例	662
练习题答案		641	11.5	Web 服务器	665
			11.5.1	Web 基础	665
			11.5.2	Web 内容	666
			11.5.3	HTTP 事务	667

11.5.4 服务动态内容	669	12.4.1 线程内存模型	696
11.6 综合: TINY Web 服务器	671	12.4.2 将变量映射到内存	697
11.7 小结	678	12.4.3 共享变量	698
参考文献说明	678	12.5 用信号量同步线程	698
家庭作业	678	12.5.1 进度图	701
练习题答案	679	12.5.2 信号量	702
第 12 章 并发编程	681	12.5.3 使用信号量来实现互斥 ...	703
12.1 基于进程的并发编程	682	12.5.4 利用信号量来调度共享 资源	704
12.1.1 基于进程的并发服务器 ...	683	12.5.5 综合: 基于预线程化的 并发服务器	708
12.1.2 进程的优劣	684	12.6 使用线程提高并行性	710
12.2 基于 I/O 多路复用的并发 编程	684	12.7 其他并发问题	716
12.2.1 基于 I/O 多路复用的并发 事件驱动服务器	686	12.7.1 线程安全	716
12.2.2 I/O 多路复用技术的优劣 ...	690	12.7.2 可重入性	717
12.3 基于线程的并发编程	691	12.7.3 在线程化的程序中使用 已存在的库函数	718
12.3.1 线程执行模型	691	12.7.4 竞争	719
12.3.2 Posix 线程	691	12.7.5 死锁	721
12.3.3 创建线程	692	12.8 小结	722
12.3.4 终止线程	693	参考文献说明	723
12.3.5 回收已终止线程的资源 ...	693	家庭作业	723
12.3.6 分离线程	694	练习题答案	726
12.3.7 初始化线程	694	附录 A 错误处理	729
12.3.8 基于线程的并发 服务器	694	参考文献	733
12.4 多线程程序中的共享变量	696		