

数据结构

1.判断 ($2^{10}=20^3$)

- 1) 若 $T(n)=a>0$, 对于 $T(n) = T\left(\frac{n}{2}\right) + O(1)$, 则不管 a 多大, 总有 $T(n) = O(\log n)$ 。
- 2) 基于 CBA 的算法对所有大小为 n 的数组时间复杂度是 $\Omega(n \log n)$
- 3) 基数排序的底层排序算法一定是稳定的
- 4) 输入随机的情况下完全二叉堆的插入平均时间复杂度是 $O(1)$, 虽然最坏是 $O(n)$
- 5) 对于任一有序列表, 即使在最坏的情况下, 折半查找的效率也不会低于顺序查找。
- 6) 即使不优化 `next[]` 数组, KMP 算法的复杂度也可以达到线性。
- 7) `fibsearch()` 时以前后黄金分割点作为轴点的常系数相同
- 8) 即使对理想随机的访问序列, 二叉伸展树也能达到均摊 $O(\log n)$ 的访问时间。
- 9) PFC(最优前缀编码)互换不同深度节点位置一定不会是 pfc
- 10) 对长度为 $m=4k+3$ 素数的散列表双平方探测一定能访问其全部元素

2.选择 ($3^8=24^3$)

- 1) 就地算法是指空间 $T(n)=$ ()
A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n \log n)$
- 2) 对于逆波兰式 $0!1+23!4+^{\wedge}56!7*8!/?/-9+$ 的值等于 2017, 则 ? 处的运算符为
A. 加号 B. 减号 C. 乘号 D. 除号 E. 乘方 F. 阶乘
- 3) 对于长度为 m 的随机 0-1 串进行串匹配时好后缀数组中 $gs[0]=1$ 的概率为
A. $\frac{1}{m}$ B. $\frac{1}{2^{m-1}}$ C. $\frac{1}{2^m}$ D. $\frac{1}{2^{m+1}}$
- 4) 一个右侧路径长度为 k 的左式堆, 其顶点数量 () 为 ()
A. 至少; 2^k B. 至少; $2^{k+1}-1$ C. 至多; 2^k D. 至多; $2^{k+1}-1$
- 5) 对于同一个长度为 n 的序列分别按照递增和递减的顺序构造 AVL 树, 那么“存在正整数 k , 使 $n=2^k-1$ ”是“两次构造的堆相同”的 ()
A. 充分不必要条件 B. 必要不充分条件 C. 充分必要条件 D. 既不充分也不必要条件
- 6) 一个具有 2017 个节点的 7 阶 B 树, 若根节点常驻内存, 则一次查找最多进行 () 次 I/O 操作
A. 7 B. 6 C. 5 D. 4
- 7) 散列表长为 2017, 采用单平方探测, 已经存入 1000 个元素, 问此时最多有 () 个懒惰删除的桶单元
A. 8 B. 9 C. 1016 D. 1017
- 8) 非法表达式 $(12)3+!4*+5$, 执行 evaluate 算法后的结果
A. 99 B. 89 C. 88 D. 98

3. 单峰向量 (好像是 16^3)

单峰向量定义为 $A[0, n)$, 其中前缀 $\{a_0, a_1, \dots, a_k\}$ 严格递增, 后缀 $\{a_{k+1}, a_{k+2}, \dots, a_{n-1}\}$ 严格递减。

- 1) 设计算法在 $O(\log n)$ 的时间内找到最大值所在位置 k 。
- 2) 证明你算法的正确性。
- 3) 证明即使在最坏的情况下, 你的算法复杂度也不会超过 $O(\log n)$ 。

4. 最大子序列和 (好像是 10')

求一个向量 $A[0, n)$ 中和最大的子序列。

- 1) 说明你的算法。
- 2) 使用伪代码描述你的算法。
- 3) 分析你算法的空间和时间复杂度。

(注: 你的得分取决于你的算法性能, 性能在 $O(n^2)$ 以内的才能得分, 时间复杂度 $O(n)$ 的才有可能得满分, 蛮力算法不得分 (因为其时间复杂度为 $O(n^3)$))

组成原理

1. 判断

- 1) 提高 cpu 主频可以加快程序执行速度, CPU 的主频越高, 指令执行的越快
- 2) 相对程序控制方式, dma 可以提高硬盘到内存的装入速度
- 3) raid6 坏两个磁盘也可以工作
- 4) c 语言 `int x,y` 若 $x > y$, 则一定有 $-x < -y$
- 5) 内存逻辑地址连续的, 物理地址不一定连续

2. 填空

- 1) -2017 的 32 位补码表示_____ (16 进制或 2 进制)。
- 2) -2017 的 IEEE 单精度浮点表示_____。
- 3) Cache 和主存的的几种映射方式_____, _____, _____。
- 4) 处理机_____逻辑电路进行算术运算, _____逻辑电路用于数据暂存, _____逻辑电路用于分支选择。

3. 选择

- 1) 以下关于五段流水线的处理机说法错误的是
A. 多个处理器不会发生结构冲突
B. 每个周期执行一个功能
C. 可以采用微程序或者硬连线设计
D. 不同的指令执行时间相同
- 2) 以下说法正确的是
A. 缓存越大程序执行速度越快
B. TLB 也是一种缓存数据和指令的缓存器
C. 指令和数据采用不同的缓存可以提高流水线速度。
D.
- 3) 以下哪个不是响应异常的处理
A. 保存 pc B. 保存通用寄存器 C. 保存异常原因 D. 恢复 pc
- 4) 以下哪种不可以解决数据冲突
A. 暂停流水线 B. 分支预测 C. 调整指令顺序 D. 数据旁路

指令可以分为 5 个阶段完成, 流水线模块延迟 10ns, 流水线寄存器延迟 5ns, 执行下面一组指令所需时间至少是多少?

```
LW R1 (0)R2
SUB R3 R1 R4
```

```
ADD R5 R1 R6
OR R7 R1 R8
ADD R9 R1 R10
```

操作系统

一 . 填空 (0.5'*10=5')

1. 若子进程执行 `exit()` 退出, 若未检测到父进程执行 `wait()`, 则子进程进入_____状态
2. 优先级反置指的是_____抢占了_____的资源, _____时低优先级进程能动态改变优先级
3. 进程调度算法中, 高响应比调度中的“响应比”的分母是程序的_____, 分子是_____
4. _____支持暂时放弃互斥资源访问权, 等待信号
5. _____提供了一个执行环境, 其中只能同时执行一个线程

二 . 判断 (0.5'*10=5')

- 1) 管程就是一个黑箱子, 程序员往里面扔函数, 同一时间只有一个函数在执行
- 2) Buddy 算法中, 释放一个空间后可以根据起始长度和大小与相邻空闲空间合并
- 3) 如果用户强制使用任务管理器 kill 一个进程, 那么即使它处于就绪状态/阻塞状态, 操作系统也要把它变成运行状态
- 4) 操作系统采用 copy on write 机制时, `fork()` 函数不会复制进程的页目录表
- 5) 管程和信号量在功能上等价
- 6) `switch_to` 函数可以直接切换, 意味着函数调用时不必保存断点。
- 7) 使用自旋锁不能保证进程按先来后到的顺序使用 cpu 资源
- 8) 管程和信号量在功能上等价
- 9) 管程将资源抽象成条件变量, 通过变量值的增减来控制进程的访问

三 . OPT、FIFO、LRU、CLOCK、LFU 这些页面置换算法, 哪些可能出现 Belady 现象? 可能的举例, 不可能的证明。(6')

四 . 阅读下面 ucore 代码, 根据代码回答相应问题

```
124/* *
125 * struct Page - Page descriptor structures. Each Page describes one
126 * physical page. In kern/mm/pmm.h, you can find lots of useful functions
127 * that convert Page to other data types, such as physical address.
128 * */
129struct Page {
130    int ref; // page frame's reference counter
131    uint32_t flags; // array of flags that describe the status of the page frame
132    unsigned int property; // used in buddy system, stores the order (the X in 2^X) of the continuous memory block
133    int zone_num; // used in buddy system, the No. of zone which the page belongs to
134    list_entry_t page_link; // free list link
135    list_entry_t pra_page_link; // used for pra (page replace algorithm)
136    uintptr_t pra_vaddr; // used for pra (page replace algorithm)
137};
```

```

// convert list entry to page
#define le2page(le, member) \
    to_struct((le), struct Page, member)

68
69/*
70 * to_struct - get the struct from a ptr
71 * @ptr: a struct pointer of member
72 * @type: the type of the struct this is embedded in
73 * @member: the name of the member within the struct
74 */
75#define to_struct(ptr, type, member) \
76    ((type *)((char *) (ptr) - offsetof(type, member)))

17struct list_entry {
18    struct list_entry *prev, *next;
19 };
20
21 typedef struct list_entry list_entry_t;
22
23 static inline void list_init(list_entry_t *elm) __attribute__((always_inline));
24 static inline void list_add(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
25 static inline void list_add_before(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
26 static inline void list_add_after(list_entry_t *listelm, list_entry_t *elm) __attribute__((always_inline));
27 static inline void list_del(list_entry_t *listelm) __attribute__((always_inline));
28 static inline void list_del_init(list_entry_t *listelm) __attribute__((always_inline));
29 static inline bool list_empty(list_entry_t *list) __attribute__((always_inline));
30 static inline list_entry_t *list_next(list_entry_t *listelm) __attribute__((always_inline));
31 static inline list_entry_t *list_prev(list_entry_t *listelm) __attribute__((always_inline));
32
33 static inline void __list_add(list_entry_t *elm, list_entry_t *prev, list_entry_t *next) __attribute__((always_inline));
34 static inline void __list_del(list_entry_t *prev, list_entry_t *next) __attribute__((always_inline));

```

le2page(*page,page_link)语句都需要展开那些宏定义？说明这个语句的含义

五 . (4') 哲学家用餐问题。题目给了一段代码，其中有用的一段是

mutex //信号量，初值为 1

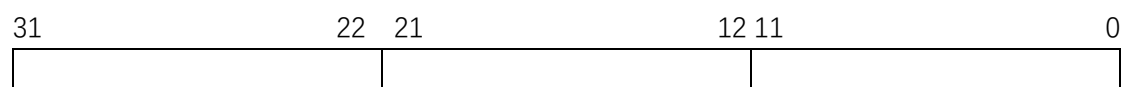
```

while {
    think();
    P(mutex);
    P(左边的叉子);
    P(右边的叉子);
    eat();
    V(左边的叉子);
    V(右边的叉子);
    V(mutex);
}

```

1. 该算法是否会导致死锁，为什么？
2. 该算法是否允许两名哲学家同时用餐，为什么？

六 . Intel X86-32 CPU 使用分页管理，每页 4KB，逻辑地址格式如图所示，这种 CPU 最多支持 4GB 内存。为了使系统能够使用 64GB 内存，使用物理内存扩展(PAE)技术，使地址长度变为 64 位，页面大小仍为 4KB。试设计逻辑地址格式，使其可以在支持物理内存扩展技术的 X86-32 CPU 上运行。(4')



10bits
page category

10bits
page table

12bits
page size

计算机网络

一. 选择 (1'*6=6')

1. 适用于奈奎斯特定理的是

I. 光纤 II. 同轴电缆 III. 红外线

A. I 和 II B. II 和 III C. I 和 III D. I、II、III

2. 使用 3000 公里长的同轴电缆, 传播速度 6m/s, 传输 64 字节的数据, 使用后退 n 帧协议, 带宽为 1.544Mbps, 为了使传输率尽量大, 序号应选 () 位。

3. 数据链路层使用的单位是

A. 比特 B. 报文 C. 帧 D. 分组

4. 选择重传协议中, 序号为 0~7, 若发送窗口大小为 7, 为了使传输不出错, 接受窗口最大为

A. 4 B. 5 C. 7 D. 8

5. TCP/IP 与 OSI

A.

B. OSI 从上到下依次是应用层, 会话层, 表示层, 网际层, 网络层, 数据链路层, 物理层

C. TCP/IP 从上到下依次是应用层, 网络层, 数据链路层, 物理层

D. TCP/IP 适用场合比 OSI 更广

6. dns 相关问题

A. 天猫双 12 购物, 不同地方两个人访问淘宝得到的 ip 一定相同

B. 用户访问 TSINGHUA.COM 和 tsinghua.com 是一样的

C. 存储 ip 是五元组

D. 数据库集中存储

二. 网桥题, 填转发表, 和王道上的一道原题几乎一样。(4')

三. (10') 两个路由器 R1、R2, R1 的 e0 端口链接着局域网 LAN1, R2 的 e0 端口链接着局域网 LAN2, R1 的 e1 端口连接着 R2 的 e1 端口。主机 A 在 LAN1 内, 主机 B 在 LAN2 内。

A~R1 之间的网段最多容纳的帧长一千多 B, 包括 12B 的头部, R1~R2 的网段最多 512B, 包括 12B 的头部, R2~B 的网段最多容纳 912B, 包括 12B 的头部。

1. 现在有 IP 地址 161.111.1.0/24 分给这些网, 包括 R1 和 R2 的两个端口, 问该如何分配, 才能使 LAN1 和 LAN2 所获得的 IP 地址数量之和最多, 写出 LAN1、LAN2 的 IP 地址范围, R1、R2 的端口地址以及它们的子网掩码。(4')

2. 若 A 要发送一个数据段 900B, TCP 头部 20B 的报文, 在网络层加了一个 20B 长的 IP 分组头部, Identification 的值为 X, 问这个 IP 分组在 A~R1, R1~R2, R2~B 上传输时, 分组的 Total length、Identification、DF、MF、fragment offset 的值各是多少?(4')

3. 若从 A 到 B 所需往返传输时间为 RTT, 现在 A 要向 B 传输 7 个 TCP 报文, 那么从开始建立连接到 A 收到最后一个确认帧结束共经历了多少 RTT?