8.

执行结果

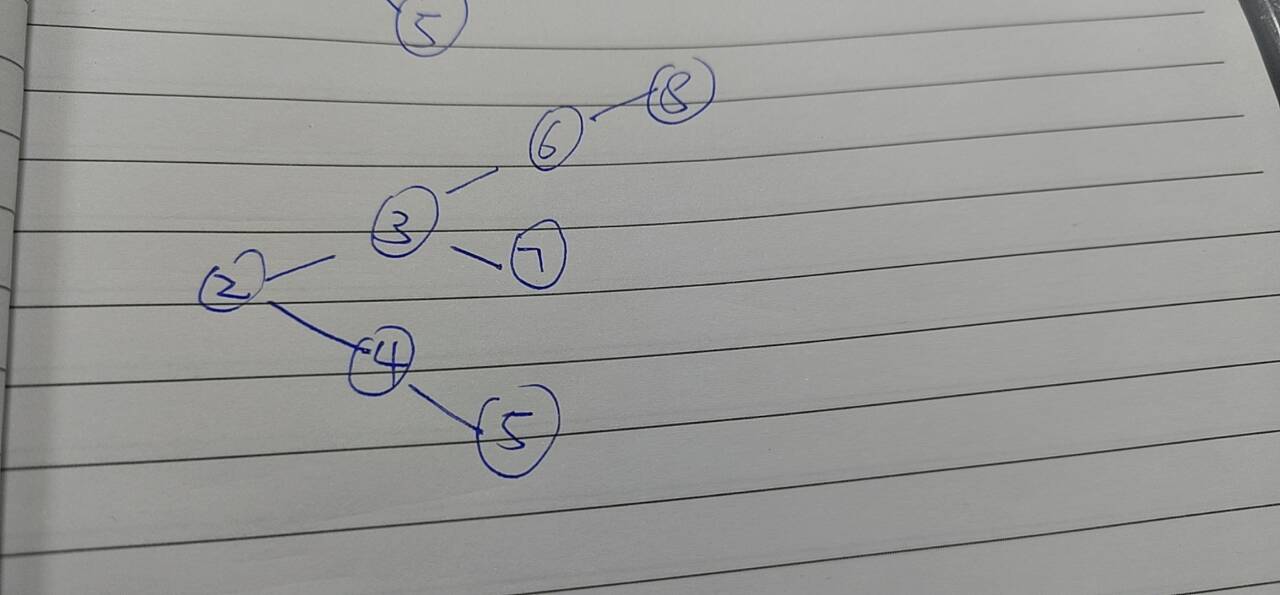
Level 2

Level 4

Level 2

Level 3

Level 2



9.

CHILD: 0

CHILD: -1

CHILD: -2

CHILD: -3

CHILD: -4

PARENT: 0

PARENT: 0

PARENT: 1

PARENT: 2

PARENT: 3

PARENT: 4

Wait函数的作用：阻塞父进程，让父进程等待其子进程结束

13.

#include <stdio.h>

#include <signal.h>

#include <sys/time.h>

#include <string.h>

#include <unistd.h>

#include <stdlib.h>

volatile sig\_atomic\_t stop = 0;

void handle\_alarm(int sig) {

printf("Alarm!\n");

}

int main() {

struct sigaction sa;

memset(&sa, 0, sizeof(sa));

sa.sa\_handler = handle\_alarm;

sigaction(SIGALRM, &sa, NULL);

struct itimerval timer;

timer.it\_value.tv\_sec = 3;

timer.it\_value.tv\_usec = 0;

timer.it\_interval.tv\_sec = 3;

timer.it\_interval.tv\_usec = 0;

setitimer(ITIMER\_REAL, &timer, NULL);

char input[2];

while (1) {

if (fgets(input, sizeof(input), stdin) != NULL) {

if (input[0] == 'S' || input[0] == 's') {

printf("Stopping...\n");

timer.it\_value.tv\_sec = 0;

timer.it\_value.tv\_usec = 0;

timer.it\_interval.tv\_sec = 0;

timer.it\_interval.tv\_usec = 0;

setitimer(ITIMER\_REAL, &timer, NULL);

break;

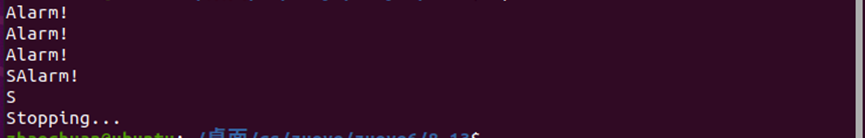
}

}

}

return 0;

}



10

6.

虚拟地址空间大小​​：32位系统虚拟地址空间为 2的32次方B

页大小​​：2KB=2的11 次方B

虚页数量：2的21次方

虚页页号：0x0030f40（需补全为32位地址 0x00030f40）

​​二进制表示​​：

00000000000000110000111101000000

​​页号（VPN）​​：取高21位（右移11位）：

000000000000001100001→十进制值=97.

​​页内偏移​​：取低11位：

11101000000→十六进制=0x740（十进制=1,856）

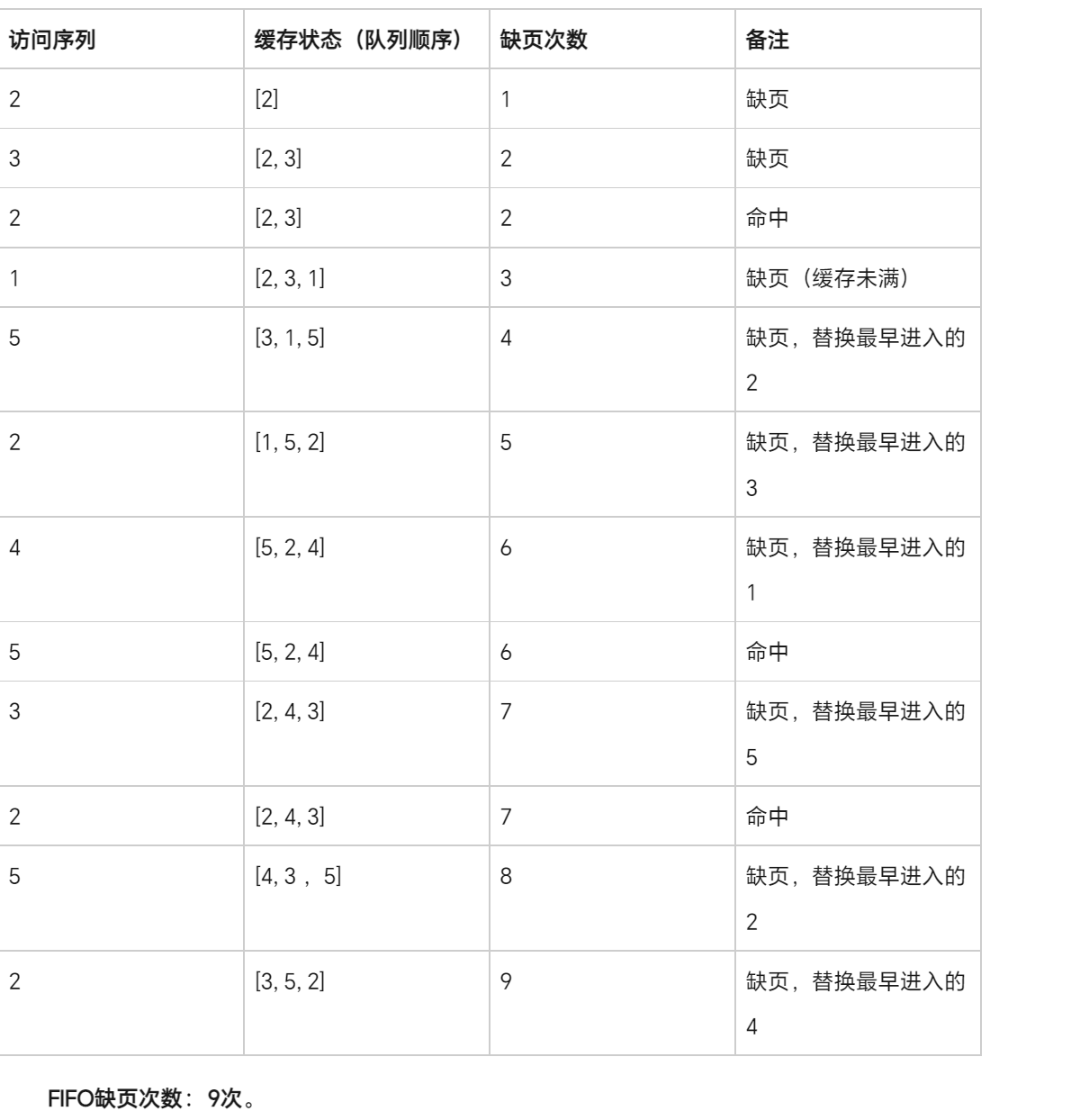
9.

选择（2）具有20GB物理主存的x86-64机器​​。

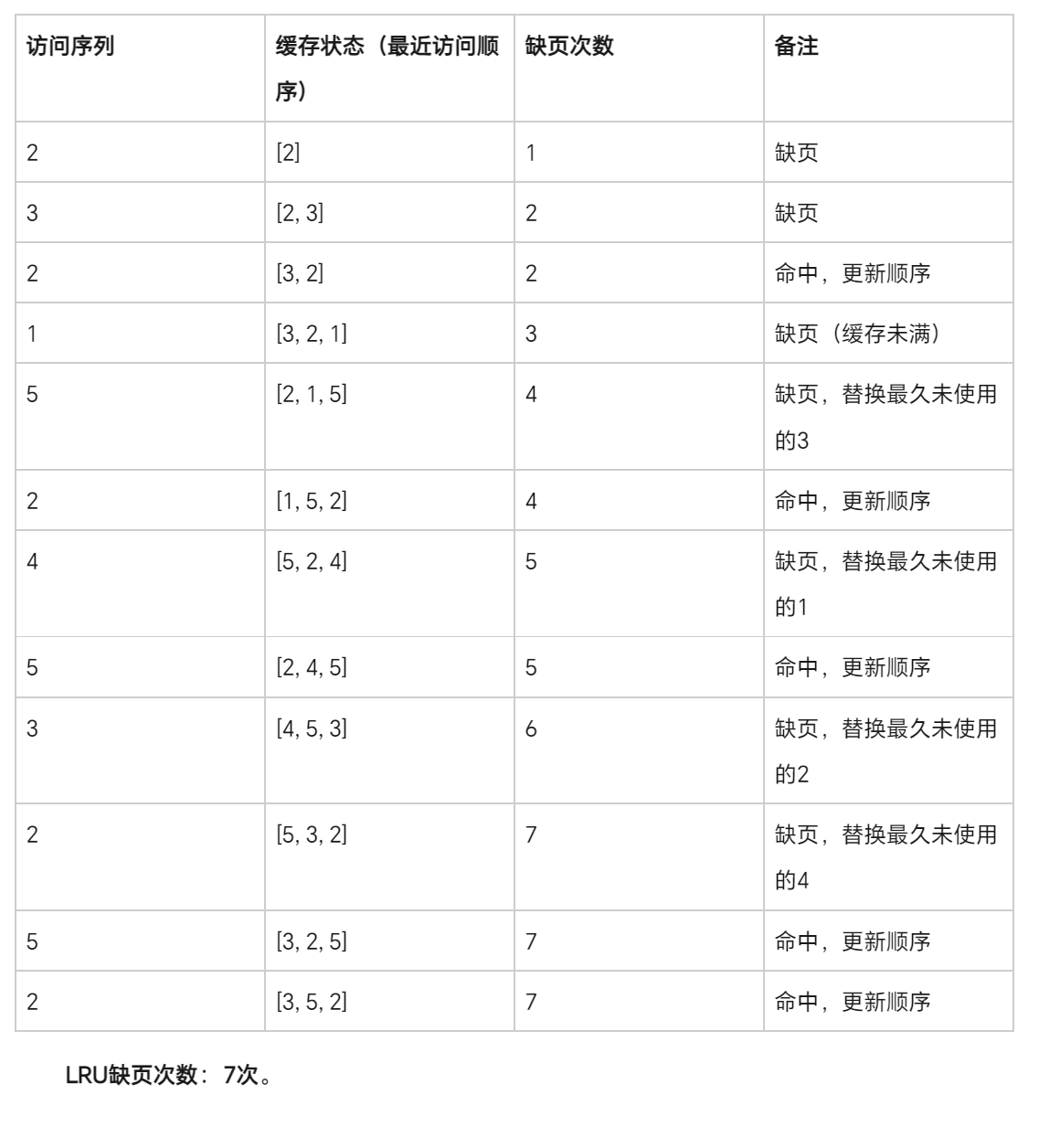
​​原因​​：（1）的PAE方案虚拟地址空间​​：32位 → 每个进程最多使用​​4GB虚拟内存​​，如果高并发大量进程共享内存，则限制较大。（2）的虚拟内存足够大，虽然物理内存较小（20GB），但可通过优化内存管理和换页机制缓解，而x86-32的4GB虚拟地址空间是硬性瓶颈，无法通过PAE解决

10.

FIFO



LRU



11

7.

copy\_array(a + 1, a, 999)​​

​​操作​​：将 a[1] 到 a[999] 复制到 a[0] 到 a[998]

copy\_array(a, a + 1, 999)​​

​​操作​​：将 a[0] 到 a[998] 复制到 a[1] 到 a[999]

原因：现代处理器采用流水线，第一个写入操作不影响后续读取，减少流水线停顿，而第二个的写入操作会影响后续的读取