# 操作系统期中试卷一 满分 100分，截止时间11月11日

同济大学计算机系 拔尖、国豪班2024 姓名 学号

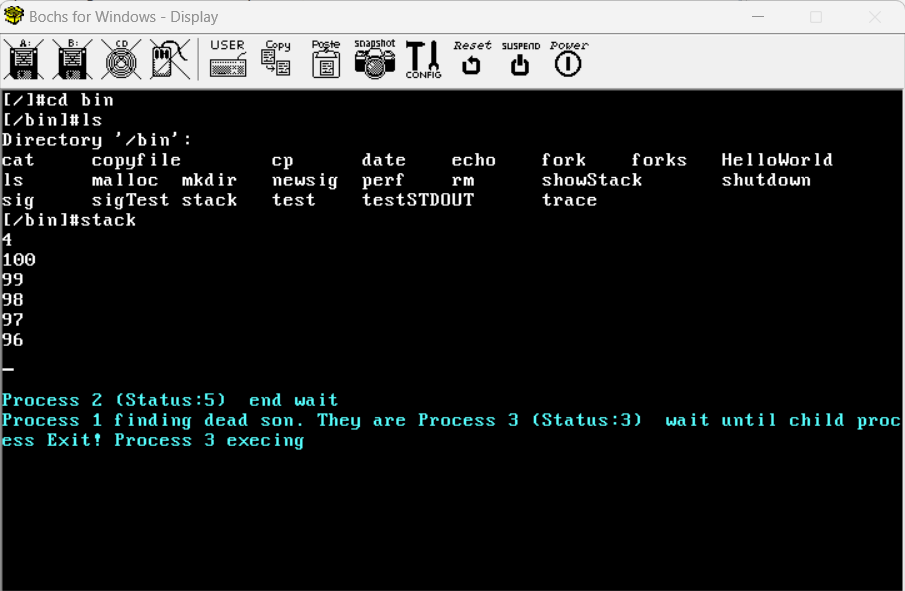
一（80分）已有经验（附件，experience-王家祺），成功去除了Unix V6++系统的相对虚实地址映射表（只读数据段还在可交换部分）。复现它。

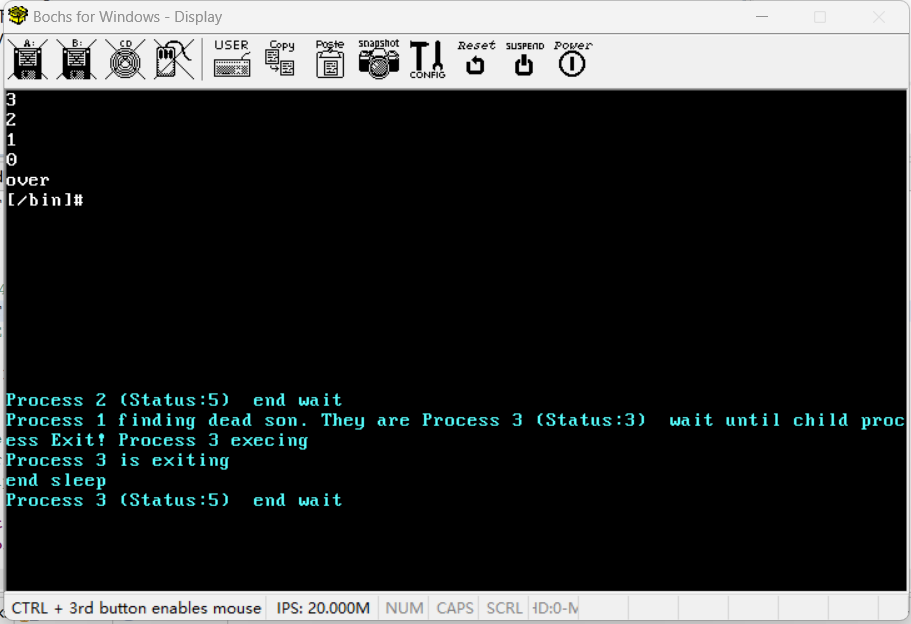
提示：（1）修改内核后，记得用tools里的工具：clean、all（2）执行系统自带的应用程序stack，能够运行成功，系统修改正确。

图形用户界面, 文本

中度可信度描述已自动生成

图 1





二（20分）二选一

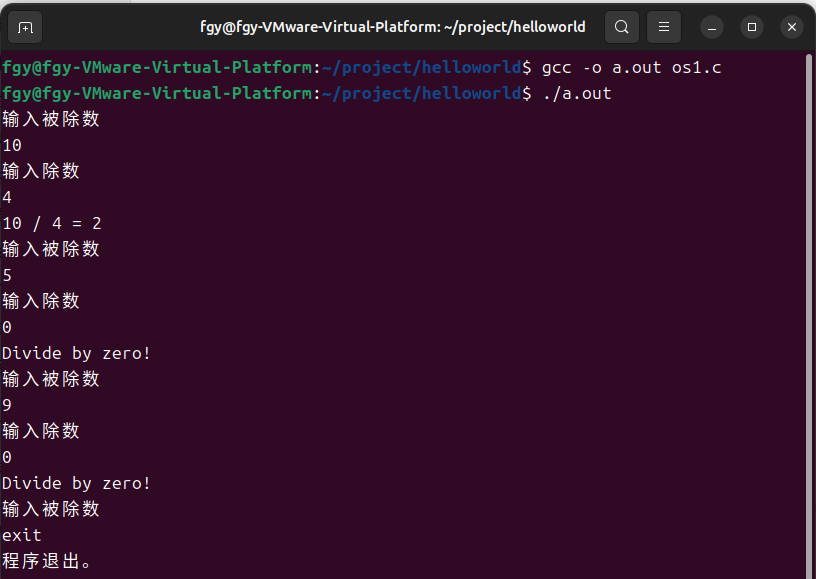
1、（20分）改进你实现的系统。在物理内存中，把只读数据段和代码段放在一起，以利共享。写出虚实地址映射关系f，实现你的映射方法。

1. (20分)
   1. （10分）Linux系统中编程实现一个专门用来执行除法操作的计算器，屏蔽除0异常。运行截图如下：（6分）



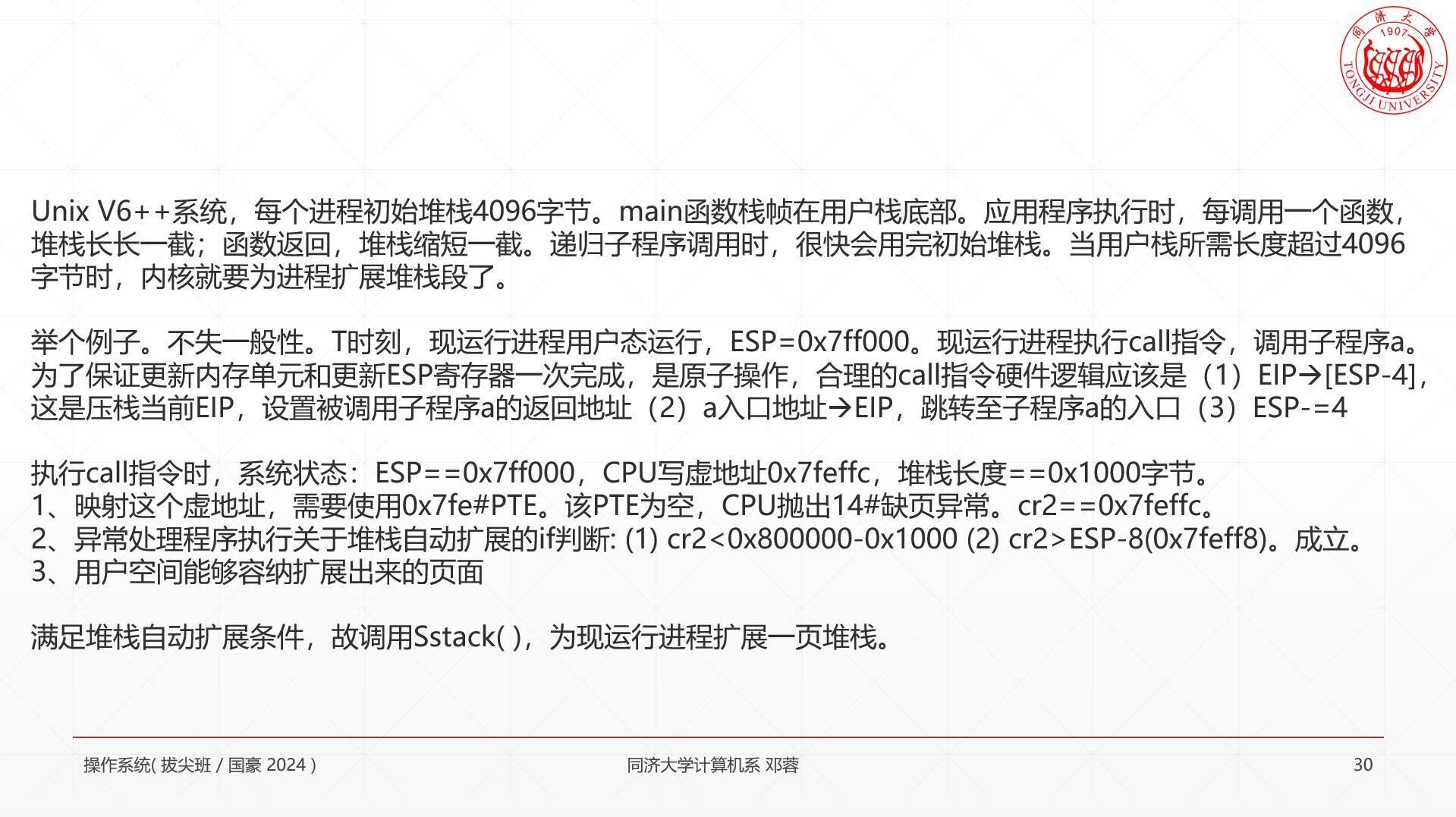
图 2

要求（1）ctrl+c杀不死这个进程，2分（2）想办法终止你改过的这个应用程序（不可以关terminal），2分



* 1. （10分）Unix V6++系统中运行应用程序stack，如图1所示。回答以下问题

(1) 以该程序的运行为背景，简介Unix V6++系统堆栈自动扩展过程。（5分）



(2) 条件

cr2 < MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize && cr2 >= context->esp – 8用来判断缺页是否因为堆栈扩展，请解释这个判断条件的合理性。（2.5分）

答：

在 cr2 < MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize && cr2 >= context->esp - 8 这个条件中：

* cr2 表示产生缺页的线性地址（虚拟地址）。
* MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize 表示当前用户空间中栈的上限地址。
* context->esp 是栈顶指针，context->esp - 8 表示当前栈顶向下扩展了 8 字节的一个范围。

**合理性解释**：

* 条件 cr2 < MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize 确保缺页地址 cr2 超出了当前已分配的栈空间，意味着栈可能需要增长。
* 条件 cr2 >= context->esp - 8 允许缺页发生在栈顶附近 8 字节的范围内，这样可以容纳一些典型的栈操作（如函数调用、参数压栈等），防止因为缺少缓冲导致栈扩展未及时发生。

因此，这个条件组合有效地判断了缺页是否由栈扩展引起，并且留有少量缓冲，以防止栈操作在正常情况下频繁触发段错误。

(3) 为什么把条件改为

cr2 < MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize && cr2 >= context->esp

stack程序运行一半会出错。（2.5分）

答：

如果改为 cr2 < MemoryDescriptor::USER\_SPACE\_SIZE - md.m\_StackSize && cr2 >= context->esp，stack 程序运行一半会出错，原因如下：

* 新条件中去掉了 - 8 缓冲，要求 cr2 的地址严格等于 context->esp 才能触发栈扩展。
* 在实际运行中，栈操作可能会将 cr2 置于 context->esp 之下的一些偏移处（例如 esp - 4 或 esp - 8），此时条件不满足，导致栈无法扩展。
* 因此，当程序需要更多栈空间进行操作时，由于未能及时扩展栈，访问非法地址会触发段错误，导致 stack 程序在运行时出错。

综上，去掉缓冲的条件使得栈的自动扩展机制无法在正常栈操作时生效，导致程序出错。

# 操作系统期中试卷二 满分 100分，截止时间11月11日

修改Unix V6++系统，严格实现时间片轮转调度。要求（1）时间1s（2）现运行进程响应中断，执行中断处理程序（3）时间片未到时，除非入睡、终止，现运行进程不会放弃CPU。

实验报告内容

1. Unix V6++并未实现严格的时间片轮转调度，是这样吗？给出判断的理由。
2. 设计和实验结果。
3. 对比Unix V6++和你的设计，进行批判性的研究。
4. 兼容并蓄、推陈出新。（这是附加分）

做好时间管理，加油同学们。

邓蓉 2024/10/28