系统安全首先是操作系统安全。

成熟的操作系统一般提供足够丰富的安全机制/功能/特性/服务等。从登录到存储加密，从内存安全保护到防病毒等，还有最重要的备份机制。// 求 20 日课上列举的 OS 安全十几条：登录认证，访问控制，文件加密，远程访问，杀毒，防火墙和 VPN，安全日志，备份，补丁更新，传输加密等等。

安卓操作系统是 linux 加上了 UI，其安全特性有针对性的变化和改进。比如没有 ROOT、VM 隔离、APK 签名、权限和交互式授权等；个人要深刻了解数据安全的坑，注意保护隐私安全等。

操作系统提供的访问控制手段可以分为两个技术路线，DAC（用户名/群组/文件）和 MAC。MAC 的代表是 selinux 和 apparmor，可以阻止合法用户/进程（文件的所有者甚至 root）做非法的（不安全的）操作，比如/etc/passwd 只有特定的用户/进程才能修改等。Windows 也提供了对重要系统文件的 MAC 保护。

新兴的操作系统比如鸿蒙需要逐步完善安全支持，包括实现一些安全和密码支持模块，提供灵活的接口，设计合理的安全体系等。

安全产品形式，主要有杀毒软件、代理服务器、入侵检测和防泄漏保护等。代理服务器是个落实安全策略的好的位置，可以记录上网、查病毒、防泄漏，甚至还能审计受 SSL 保护的密文流量。典型代理服务器是 Squid，配合使用 icap、sslbump 和 calmav 等技术执行安全管理规则。从新闻中每天的泄露案就可以明白入侵和泄露的事件频发生，很多企业和网络的防范很不给力。

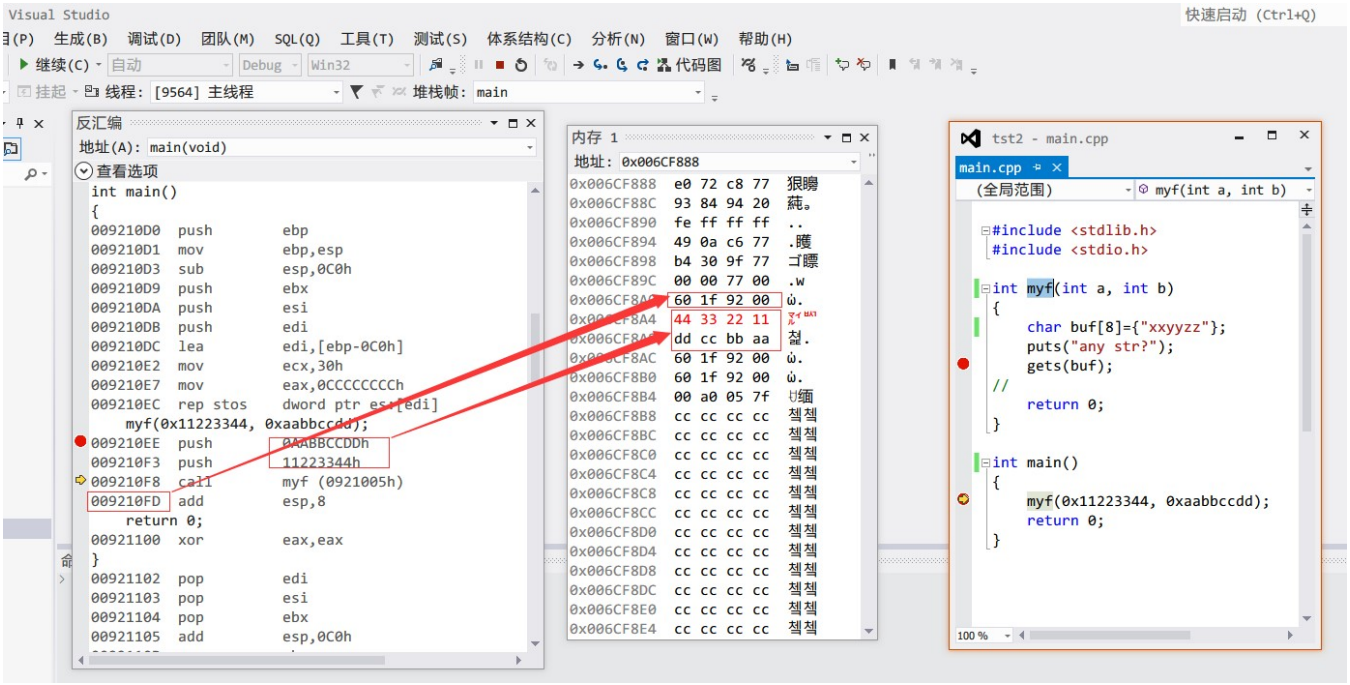
RAS（可靠性 Reliability,可用性 Availability,可服务性 Serviceability）是系统安全追求的重要指标。注意，操作系统中的优化工作，很多和 RAS 的追求是反向的。比如内存管理的优化，使用 fork 和动态库（.dll/.so）可以极大地节约内存，但是这种单个页面被多个进程共享的形式，在出现存储器比特翻转错误时会造成单一故障(single failure)现象，违反了 RAS 的安全隔离原则。

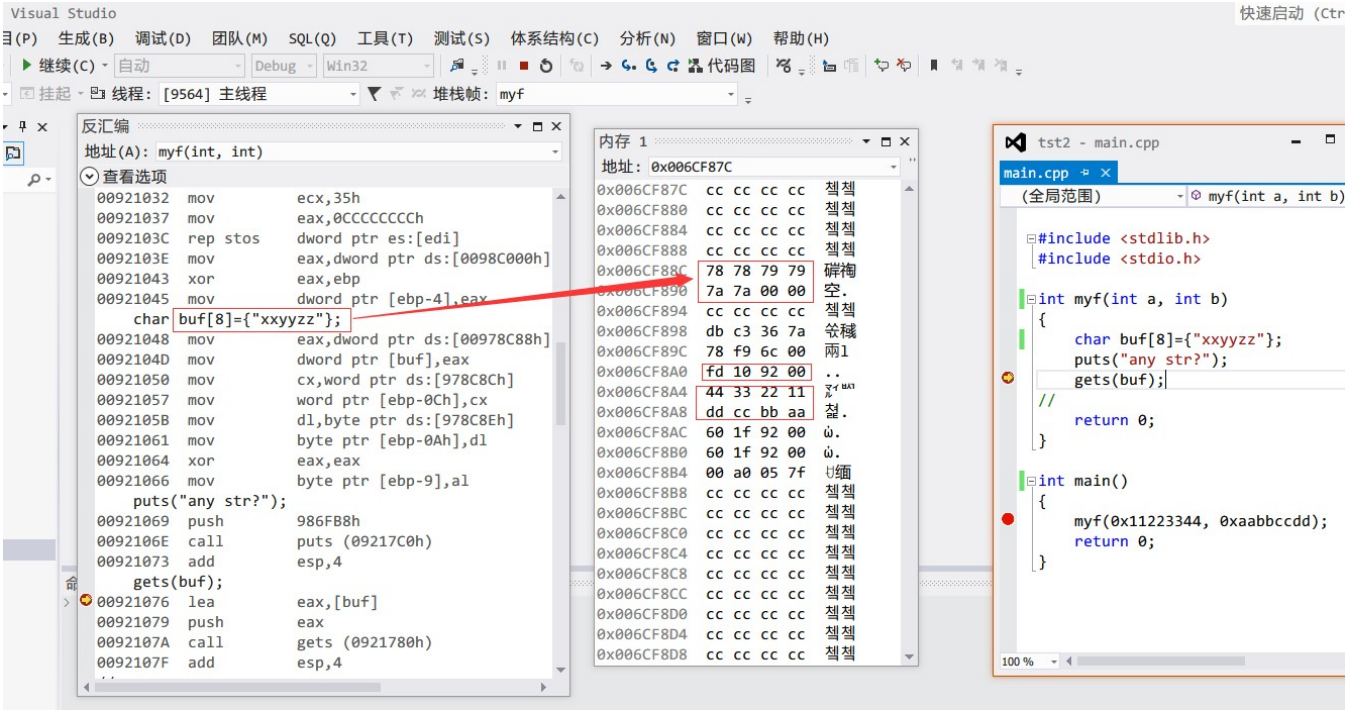
内存安全是一个重要的基础安全因素。目前各种漏洞大部分和内存不安全有关系，典型代表就是缓冲区溢出（buffer over flow）。后面附用 visual studio 展示的函数调用时参数、返回地址、临时变量等的栈中场景，可以帮助理解为什么 bof 这么普通、常见而又不安全。

计算机病毒的威胁感觉已经被很好地遏制了，主要是信息系统长期对抗病毒的努力。目前主要是入侵和勒索型病毒，终极对抗办法是科学备份。学习病毒和对抗技术，主要可以从写一个某种脚本语言的（或用伪代码）、文件寄生型的病毒入手。

密码子系统是系统安全的支撑技术之一，从加密到防病毒（比如重要的 windows 系统文件都要有微软的数字签名，因此能发现病毒等的异常修改动作）都用到密码技术。一个好的安全系统应该有一个好的密码子系统的接口，允许和方便第三方开发和注册更多的密码算法。重点关心一下对称算法和散列函数的 API 设计。

附，看两个场景，函数调用时栈内场景，一个是 32 位环境，一个是 64 位环境。可以只关心 32 位环境的。





以上 x32 位栈，还是比较容易看懂的。先入栈了参数，然后入栈了返回地址，然后在栈内给临时变量 buf 开辟了空间。

X64 的，主要变化是函数参数的前 4 个通过寄存器而不是栈传递，第 5 个才开始用栈。