# 阅读报告5

1.论文题目：

Singularity: Rethinking the Software Stack

2.阅读报告：

每个操作系统都包含一系列设计决策。即使硬件和软件已经发展，当今最流行的操作系统背后的许多决定仍然没有改变。操作系统几乎构成了所有软件栈的基础，因此现有系统中的不足之处会产生普遍影响。

Singularity是一个多年的研究项目，重点是通过系统，语言和工具领域的创新来构建可靠的系统。我们构建了一个研究操作系统原型（称为Singularity），扩展了编程语言，并开发了用于指定和验证程序行为的新技术和工具。展示了使系统更健壮和可靠的新技术以及体系结构决策的实用性。

论文描述了Singularity项目根据编程语言和验证工具的进步重新审视这些设计选择的努力。Singularity系统包含三个关键的体系结构特征：用于保护程序和系统服务的软件隔离流程，用于通信的基于合同的通道以及用于验证系统属性的基于清单的程序。

Singularity的起始点是如果底层是安全的managed语言构建系统，就不需要复杂的虚存管理，所有应用可以跑在一个地址空间。上下文切换成本也会降低，可以跟erlang一样做很多并发进程。

大多数大型操作系统的思想都来自Multics，当然那时候的机器非常低速，通过汇编语言来编写，仅通过不同用户间的分时操作来达到文件的安全保护目的。而Singularity是根据当今时代的情况，语言，编译器和工具的进步为开发软件提供了可能性。Singularity使用类型安全语言和抽象指令集来实现我们称之为软件隔离进程（SIP）的功能。SIP为操作系统进程提供了强大的隔离保证（隔离的对象空间，独立的GC，独立的运行时），而没有硬件强制保护域的开销。在目前的Singularity原型中，SIP非常便宜; 它们在内核地址空间的环0中运行。

Singularity使用这些进步来构建更可靠的系统和应用程序。例如，由于SIP创建和执行起来非常便宜，Singularity会在其自己的SIP中运行每个程序，设备驱动程序或系统扩展。不允许SIP共享内存或修改自己的代码。因此，我们可以对在SIP中运行的代码提供强大的可靠性保证。我们可以在编译或安装时验证有关SIP的更广泛的属性，而不是在传统OS进程中运行的代码所能做到的。更广泛的静态验证应用对于预测系统行为以及为用户提供有关可靠性的有力保证至关重要。