

近年来，Rust重构的热潮迅速席卷整个开发圈。作为一颗日渐兴起的新星，与其他编程语言相比，Rust在安全性、高效性和可维护性等方面都展现出明显的优势。因此，越来越多的大型项目和关键系统纷纷选择采用Rust进行代码改写和重构。



值得期待的是，统信UOS服务器版将于不久后推出Rust版Bash，名为utshell，旨在突破传统Shell所面临的安全局限，构建一个更具创新性和竞争力的操作系统。

utshell何时崭露头角？

6月初，**统信UOS服务器版V20（1060）**隆重推出，不仅带来了突破性的功能和性能提升，还集成了utshell的部分功能，为即将发布的完整版utshell提供了初步体验。

为了推动自主操作系统的生态建设，统信UOS始终拥抱开源、贡献开源。此次，统信UOS已将utshell的前期成果同步到了deepin、欧拉和龙蜥社区。

欢迎下载和体验，同时也衷心期待更多感兴趣的朋友加入我们，与我们一同建立更加繁荣的自主操作系统生态。

deepin社区地址：

https://github.com/linuxdeepin/utshell

欧拉社区地址：

https://gitee.com/openeuler/utshell

龙蜥社区地址：

https://gitee.com/anolis/utshell

utshell有何惊艳魅力？

经过Rust重构和安全补丁的引入，utshell已成为一个更安全、更强大的Shell程序，为Unix和Linux系统的用户提供了更加可靠且高效的运行环境。

有效避免命令被篡改

在Bash中，有一些命令没有对参数进行防篡改检查，因而给系统带来了安全风险。

例如，alias命令允许用户为其他命令或命令序列创建别名，然而它并未对参数进行防篡改检查，进而导致未经检查的参数可能会被利用来执行恶意代码或进行未经授权的操作。



为了应对上述问题，utshell对alias命令的规则进行了全面详细的明确界定，目的在于确保使用的便捷性，并且能够有效地防止命令被恶意篡改，从而限制恶意行为或误操作的发生。这一举措不仅可以确保系统的安全性，还能提升用户体验。

* 当别名和$PATH下可执行文件重名时，只允许别名和可执行文件一致。例如，如果我们想要创建一个别名“ls”，那么与之对应的可执行命令也必须是“ls”，且允许在命令中添加参数，以便用户可以根据自己的需求对命令进行个性化制定。
* 别名不能与其他别名、函数名称发生重名，以避免在使用别名时出现二义性，进而更好地确保操作系统能够正确理解用户的指令，避免混淆和误解。

当然，不仅仅限于alias命令，utshell还对所有命令进行了详尽检查，并对那些可能受到恶意篡改威胁的命令进行了规则的优化处理，如hash和function命令等。

深度保护参数不溢出

在Bash中，mapfile命令是用于对数组进行赋值的工具，但它并没有对数组的越界情况进行有效的检查和控制。当用户输入的参数数量超过数组容量时，溢出部分可能会被误当作命令来执行，从而引发潜在的安全隐患。



为了避免上述情况的发生，utshell采取了对数组边界的判断措施。这一举措不仅能有效避免数组越界错误的发生，还能预先制定一些安全措施。

例如，当传入的参数过多时，自动截断或给出警示等，从而保证只有合法、预期的参数才能被传递给mapfile命令。

“源生”安全，自主可控

通过使用Rust重构Bash，utshell不仅能继承Rust独特的“源生”安全性，还能在代码层面实现更高的自主可控性，从而极大地提升了系统的安全性和可靠性。

“源生”安全是将安全元素融入系统的整个生命周期中，能够有效预防安全漏洞和攻击，是保障系统安全的重要手段。

代码自主可控，意味着能够完全自主地控制和管理utshell内部的代码逻辑和执行过程，使得我们能够更加灵活地定制和优化代码，从而提升脚本的执行效果和开发体验。

Bash为何需要重构？

Bash是一种广泛应用于各种Unix和Linux系统的Shell程序，同时属于统信UOS分层分类设计方案的核心层，其安全性和稳定性至关重要。

**Tips:**

分层分类的核心思想是“**垂直分层、水平分类、数据驱动、算法先行**”，旨在实现操作系统软件供应链的自动化管理。它结合了人工智能技术，能够随着统信UOS的不断演进，对新增或更新的软件包实现自动识别，以及智能分层和分类。

使用Rust重构Bash后，不仅能够更好地满足现代化的需求和挑战，还可以最大限度地预防用户操作和软件缺陷导致的软件健壮性和用户安全问题。



* 因历史悠久，Bash的代码已经堆积了大量的复杂性。因此，进行重构是确保Bash的可靠性和可维护性的必要步骤。
* 随着技术的飞速发展和新功能需求的不断扩展，Bash的设计和架构显然有些过时。通过重构，可以使Bash更加灵活、高效，并能轻松适应现代化软硬件。
* Bash在解析和执行命令时存在一些性能上的瓶颈，这可以通过重构来优化。同时，通过重新审查代码和引入更严格的安全措施，可以减少潜在的漏洞和安全风险。
* 通过对Bash进行重构，可以使代码更加清晰、易读，并且更易于理解和维护，进而提升开发者的工作效率和用户的使用体验。

Rust因何成为首选？

作为一门系统级编程语言，Rust在性能优化、资源管理、并发处理、代码安全等多个方面，都展现出了卓越的能力，为开发者提供了更高效、更可靠的软件开发环境，同时也为操作系统的发展带来了崭新而有前景的可能性。



众所周知，传统操作系统在处理内存管理和安全性等方面经历了许多困难和挑战。然而，Rust的强大特性可以在很大程度上解决这些问题。

鉴于此，统信UOS在对Bash进行重构时选择了Rust，充分利用了Rust的内存安全特性。

统信软件一直坚持提供极致的安全防护，不仅拥有最严苛的流程管理、最极致的设计理念，更重要的是拥有最硬核的安全技术。本次推出的**「Rust版Bash」**新方案，是对统信UOS服务器版全栈系统安全防护体系的进一步完善和升级。



**统信UOS服务器**版汲取国内外主流社区技术栈优势，深入技术底层结合国内外设计标准与规范以及各类用户业务应用需求，积极开展技术创新，全面支持国内外主流CPU架构和处理器厂商，在各种应用环境中，满足**强安全、高稳定、高性能、易维护以及泛兼容**等要求，是构建信息化设施环境的基础软件产品：

◈强安全：

* 等保四级GB/T 20272-2019认证
* 商密二级GM/T 0028认证
* 全栈生态链支持国密
* 国际知名安全社区OpenSCAP官方首批支持的中国操作系统
* 21天推送一次补丁更新，每周进行漏洞修复推送

◈高稳定：

* 稳定版 Linux 4.19&5.10双内核
* 热补丁升级机制
* LTP 7\*24小时测试用例通过率超98%
* 长时高负荷运行，卸荷后60秒内实现系统及各应用流畅运行

◈高性能：

* 内存拷贝页技术，拷贝性能提升超70%，满并发优化提升10%
* 多种自研性能优化成果：文件读写性能提升4%、调度延迟减少10%、多并发锁竞争场景性能提升5%……
* 相比同类产品，UnixBench等测试全面领先

◈易维护：

* 提供自研统信有易服务器迁移软件
* 提供自研统信有幄自动化运维工具
* 提供面向场景的AI系统调优能力

◈泛兼容：

* 历史版本兼容性>98%，提供平滑升级方案
* 组件模块化设计，为用户最大限度解决底层依赖问题
* 全面通过整机适配兼容性测试（多种不同型号的整机）



**统信软件服务器操作系统与云计算产线总经理孟杰**表示，Rust重构潮流的兴起反映了开发者对先进技术的追求，并彰显了Rust的优势和潜能。统信UOS即将发布的Rust版Bash不仅仅是一次创新性的尝试，更是带来了一场操作系统领域的革新，为用户提供了更强大的安全特性、更广泛的功能选择和更出色的用户体验。除了Bash，我们还对Sudo进行了重构，敬请期待！

©统信软件技术有限公司。访问者可将本网站提供的内容或服务用于个人学习、研究或欣赏，以及其他非商业性或非盈利性用途，但同时应遵守著作权法及其他相关法律的规定，不得侵犯本网站及相关权利人的合法权利。除此以外，将本网站任何内容或服务进行转载，须备注：**该文档出自【faq.uniontech.com】统信软件知识分享平台**。否则统信软件将追究相关版权责任。