

# The Future of the Shell: UNIX and Beyond

2021-06-03 1–2:30pm ET

Michael Greenberg\*

Pomona College  
Claremont, CA, USA  
michael.greenberg@pomona.edu

Konstantinos Kallas\*

University of Pennsylvania  
Philadelphia, PA, USA  
kallas@seas.upenn.edu

Nikos Vasilakis\*

MIT  
Cambridge, MA, USA  
nikos@vasilak.is

## ABSTRACT

The UNIX shell is fifty years old, and it continues to be the primary way to configure, deploy, and manage systems of all kinds. What do the next fifty years hold? What is the command-line interface of the 21<sup>st</sup> century?

This 90-minute panel brings together researchers and engineers from disparate communities (systems, languages, security) to think about the shell’s strengths and weaknesses, challenges and opportunities around the shell, and the shell’s future.

### ACM Reference Format:

Michael Greenberg, Konstantinos Kallas, and Nikos Vasilakis. 2021. The Future of the Shell: UNIX and Beyond: 2021-06-03 1–2:30pm ET. In *Workshop on Hot Topics in Operating Systems (HotOS ’21), June 1–June 3, 2021, Ann Arbor, MI, USA*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3458336.3465296>

The shell is ubiquitous. It is an important system interface that developers and system administrators use for interacting with their systems. It is also the primary, default choice for scripting system orchestration, automation, and maintenance. Finally, it is used for quick prototyping of more complex solutions, especially in domains like data processing, data ingestion, and data cleaning. The shell’s composition primitives let scripts combine tools written in a variety of languages, making it an exemplar of extensibility.

The shell is also a confusing and dangerous object of disgust. There is little in the way of support, and best practices for shell scripting are not well known. The shell has been around for fifty years, but it has only just begun to enjoy the benefits of fifty years of advances in technology and tooling.

\*Alphabetical order.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*HotOS ’21, June 1–June 3, 2021, Ann Arbor, MI, USA*

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8438-4/21/05.

<https://doi.org/10.1145/3458336.3465296>

**Renewed Interest in the Shell.** A combination of trends in containerization, data science, programming-language techniques, and the cloud have brought shell scripting back to prominence [1–10]: What does POSIX compliance really mean in the context of the shell [5]? What do package install scripts actually do [1]? Can we offload shell computations efficiently in the cloud [7]? How can we parallelize UNIX pipelines with minimal-to-zero user involvement [9]? Recent work addressed these long-standing questions—but a lot more work is underway, signaling (1) renewed interest in the classic shell and alternative variants, and (2) the emergence of a community thinking about its future applications.

**Format.** The 90-minute event (on June 3<sup>rd</sup> 2021 from 1–2:30pm ET) has a small panel of experts on shell-related topics. The event itself is discussion-based, with the panelists providing single-page statements of their interest in and vision for the shell, including one or more *challenge problems*, which will be distributed to participants in advance.

Introduction	10min
Panelists summarize statement	10min
Breakout in three topic-oriented rooms	30min
<i>Break</i>	5min
More breakout, attendees shuffle rooms	15min
Panelists summarize discussions for all attendees	15min
Closing	5min

The organizers will divide themselves among the breakout rooms to help record conversations.

**Panel** The panelists are Arjun Guha (Northeastern), Deepti Raghavan (Stanford), Chet Ramey (Case Western Reserve), and Diomidis Spinellis (AUEB; TU Delft).

**Discussion topics** The precise topic rooms will be determined by surveying panelists and participants. We anticipate two primary possibilities for choosing topics: following disciplinary distinctions (systems, languages, HCI) or following crosscutting concerns (safety, performance, expressivity).

**Final report** In collaboration with attendees, the organizers will produce a summary document of the panel, to be made available at <https://fut-shell.github.io/>.

## REFERENCES

- [1] Benedikt Becker, Nicolas Jeannerod, Claude Marché, Yann Régis-Gianas, Mihaela Sighireanu, and Ralf Treinen. 2020. Analysing installation scenarios of Debian packages. In *Tools and Algorithms for the Construction and Analysis of Systems*, Armin Biere and David Parker (Eds.). Springer International Publishing, Cham, 235–253.
- [2] Ting Dai, Alexei Karve, Grzegorz Koper, and Sai Zeng. 2020. Automatically Detecting Risky Scripts in Infrastructure Code. In *Proceedings of the 11th ACM Symposium on Cloud Computing* (Virtual Event, USA) (*SoCC '20*). Association for Computing Machinery, New York, NY, USA, 358–371. <https://doi.org/10.1145/3419111.3421303>
- [3] Michael Greenberg. 2018. The POSIX shell is an interactive DSL for concurrency. <https://cs.pomona.edu/~michael/papers/dsldi2018.pdf>.
- [4] Michael Greenberg. 2018. Word Expansion Supports POSIX Shell Interactivity. In *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming* (Nice, France) (*Programming'18 Companion*). Association for Computing Machinery, New York, NY, USA, 153–160. <https://doi.org/10.1145/3191697.3214336>
- [5] Michael Greenberg and Austin J. Blatt. 2020. Executable Formal Semantics for the POSIX Shell: Smoosh: the Symbolic, Mechanized, Observable, Operational Shell. *Proc. ACM Program. Lang.* 4, POPL, Article 43 (Jan. 2020), 31 pages. <https://doi.org/10.1145/3371111>
- [6] Shivam Handa, Konstantinos Kallas, Nikos Vasilakis, and Martin Rinard. 2020. An Order-aware Dataflow Model for Extracting Shell Script Parallelism. *arXiv preprint arXiv:2012.15422* (2020).
- [7] Deepti Raghavan, Sadjad Fouladi, Philip Levis, and Matei Zaharia. 2020. POSH: A Data-Aware Shell. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, 617–631. <https://www.usenix.org/conference/atc20/presentation/raghavan>
- [8] Diomidis Spinellis and Marios Fragkoulis. 2017. Extending Unix Pipelines to DAGs. *IEEE Trans. Comput.* 66, 9 (2017), 1547–1561.
- [9] Nikos Vasilakis, Konstantinos Kallas, Konstantinos Mamouras, Achilles Benetopoulos, and Lazar Cvetković. 2021. PaSh: Light-Touch Data-Parallel Shell Processing. In *Proceedings of the Sixteenth European Conference on Computer Systems* (Online Event, United Kingdom) (*EuroSys '21*). Association for Computing Machinery, New York, NY, USA, 49–66. <https://doi.org/10.1145/3447786.3456228>
- [10] Keith Winstein and Hari Balakrishnan. 2012. Mosh: An interactive remote shell for mobile clients. In *2012 USENIX Annual Technical Conference (USENIX ATC 12)*. 177–182.