Unit 7: Introduction to Operating Systems

Required Reading

Saltzer, J. & Schroeder, M. (1975) The Protection of Information in Computer Systems. Proceedings of the IEEE 63(9): 1278-1308.

Summary
The document "The Protection of Information in Computer Systems" provides an in-depth exploration of mechanisms to safeguard digital information from unauthorized access, modification, and disruption. It focuses on architectural structures, both hardware and software, necessary to achieve robust information protection. The paper is structured into three main sections: the first introduces fundamental concepts, desired functions, and basic security mechanisms; the second delves into advanced protection architectures such as capability systems and access control lists; and the third reviews the state of the art and ongoing research in the field. Various security principles, including complete mediation, least privilege, and fail-safe defaults, are discussed to guide the development of secure systems. The paper also presents a glossary of key terms related to cybersecurity, emphasizing authentication, authorization, and encryption techniques.
Reflection
Reflecting on the document, it offers valuable insights into the complexities of securing computer systems, underscoring the challenges posed by evolving threats and the necessity of a layered security approach. The emphasis on principles such as economy of mechanism and separation of privilege highlights the importance of designing systems that are both secure and practical to implement. The document's exploration of different protection models, from simple all-or-nothing systems to sophisticated user-defined sharing controls, demonstrates the diverse needs of various computing environments. However, while the paper provides a thorough technical foundation, the rapid advancement of cyber threats since its publication suggests that these principles must continually evolve to address modern challenges such as cloud security and artificial intelligence-driven attacks.

Szabo, G. (2018) Create your own interactive shell with cmd in Python.

Summary
The document "Create Your Own Interactive Shell with cmd in Python" provides a step-by-step guide to building a command-line interface (CLI) using Python's built-in cmd module. It starts with creating a basic interactive shell by subclassing the Cmd class and running a command loop. The guide then explores adding functionality by implementing do_* methods for specific commands, customizing the prompt and banner, and handling default actions when an unknown command is entered. The document also covers documentation of commands using help_* methods or docstrings and addresses handling standard termination signals like Ctrl-D with custom exit behavior. A complete example integrating all discussed concepts is provided at the end.

Reflection
Reflecting on the document, it offers a practical and concise introduction to building interactive shells, making it an excellent resource for developers looking to enhance their applications with a CLI. The step-by-step breakdown, along with sample code snippets, provides clarity and ease of implementation. The ability to customize commands, prompts, and error handling makes the cmd module a versatile tool for developing user-friendly command-line applications. However, while the guide covers essential functionality, it could further explore more advanced topics such as integrating with external systems or adding authentication layers. Overall, the document effectively equips readers with the fundamental skills to create and expand their own interactive shells.

Praka, D. (2018) Write a shell in Python.

Page does not exist

Arnaut, W., Oliveira, K. & Lima, F. (2010) OWL-SOA: A Service Oriented Architecture Ontology Useful during Development Time and Independent from Implementation Time, IEEE.

Summary
The document introduces OWL-SOA, a Service-Oriented Architecture (SOA) ontology designed for development time, independent of specific implementation technologies. Unlike existing frameworks like OWL-S and WSMO, which focus on runtime and are constrained to web services, OWL-SOA addresses a broader scope. The ontology facilitates the organization and retrieval of services by incorporating development artifacts, business processes, and diverse technologies. Created through the integration of OWL-S and WSMO using methodologies like SABIO and PROMPT, OWL-SOA includes extensions for supporting multiple technologies and offers a semantic foundation for SOA repositories. This ontology was validated using services from a multinational organization's repository.
Reflection
The development of OWL-SOA underscores the significance of adaptability and interoperability in the modern digital landscape. By bridging the gaps left by earlier ontologies, it paves the way for more efficient service reuse and integration during development. The use of structured methodologies ensures a robust foundation, and its independence from specific technologies highlights its versatility. The study emphasizes the growing need for semantic tools that align with diverse implementation contexts, signaling a shift towards more inclusive and flexible system architectures. It inspires further exploration of ontologies tailored for evolving business and technical requirements.

Additional Reading

Silberschatz, A., Galvin, P. & Gagne, G. (2018) Operating System Concepts. 10th ed. Hoboken, N.J: Wiley.

- Chapters 2-8, 16, 18 and 19

Garfinkel, S., Weise, D. & Strassmann, S., (1994) The Unix-Haters Handbook. 1st ed. San Mateo, Ca.: IDG Books Worldwide.

Corbato F. & Vyssotsky V. (1965) Introduction and Overview of the Multics System. Proceedings of the Joint Computer Conference, ACM 1:185– 196.