

# Unikernel Linux (UKL)

Jakob Schmid

## Abstract

This report discusses Unikernel Linux (UKL), an approach to introduce a unikernel target into the Linux kernel. The specialized demand of cloud services has recently given rise to a resurgence of library operating systems in the form of unikernels. The authors of the UKL paper want to show that the Linux kernel can be modified to include the benefits of unikernels, while maintaining the ecosystem of applications and maintainers of Linux.

## 1 Introduction

Modern cloud services are often highly specialised, to point of a microservice, where an applications is split up into a collection of loosely coupled services, that communicate through lightweight protocols and each only fulfill a single purpose. These services share a few key demands. They need to be able to communicate efficiently with each other. Since they need to communicate with other services, they are exposed to the internet to some degree and thus should be as secure as possible. Due to the single purpose nature of the microservices they often execute as a single process. Lastly the image the service is executed with and the memory usage should be as small as possible. The services are often executed in a virtual machine on rented hardware, so lower requirements in memory and storage can improve the profitability of a service.

This has caused a resurgence of research exploring the concept of a library OS and the emergence of unikernels. In a library OS a target application is linked with a set of libraries that provide all the services a regular OS would usually provide. The resulting executable can then be deployed directly to hardware.

## 2 Background

```
/* code snippet */
while (!sleep)
    sleep++;
```

Listing 1: A sample code snippet

Here you can include a sample figure. Use something like

```
\includegraphics[scale=.8]{template}
```

to include an encapsulated postscript figure. The *scale* argument can be used for scaling the picture, although it may scale the font incorrectly.

Figure 1: Sample Figure

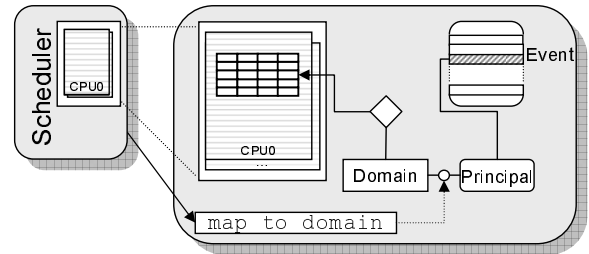


Figure 2: Sample figure automatically from Windows prn.

## 3 Related Work

Works [1] and [2] are relevant but different.

## 4 Approach

## 5 Conclusion

## References

- [1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of the 19th Symposium on Operating System Principles*, pages 164–177, Bolton Landing, NY, Oct. 19–22 2003.
- [2] I. Pratt, K. Fraser, S. Hand, C. Limpach, A. Warfield, D. Magenheimer, J. Nakajima, and A. Malick. Xen 3.0 and the art of virtualization. In *Proceedings of the 2005 Ottawa Linux Symposium*, pages 65–85, Ottawa, Canada, July 2005.