ARBEITSPROBEN

ZUKÜNFTIGE PROJEKTE

- Beschäftigung in einem Themengebiet, welches mich interessiert und mir auch hilft, die Erkenntnisse anzueignen, die mich auch bei der Verwirklichung meiner Interessen unterstützen.

- Recherche, ob ein an mach_vm_map angelehnter Linux process_vm_map System Call mainline-fähig wäre.
- JIT-Debugging auf Linux ermöglichen, wie es unter Windows und Mach schon der Fall ist.
- Microsoft Maren (Windows Eingabemethode, welches lateinische Buchstaben zu Arabisch transliteriert) auf Wine portieren (Eingabemethode an sich schon implementiert[1]).

LAUFENDE PROJEKTE

> *Es sollte nicht der Fall sein, dass man einen eigenen Treiber für von Linux bereits unterstützte Geräte schreiben muss, nur damit diese jeweils Echtzeitethernet unterstützen.*
>
> — Dr. Andreas Bihlmaier, robodev GmbH

Im Rahmen der Bachelorarbeit wurde das Linux Networking-Subsystem untersucht und ein generischer Treiberadapter entworfen und implementiert[2], der etablierte Linux Switching-APIs wiederverwendet, um openPOWERLINK im Kernel, ohne Reimplementierung der Treiber, zu fahren. Erste Tests weisen mit den mitgelieferten Treibern vergleichbares Echtzeitverhalten auf.

[1] https://github.com/a3f/Bowdlator
[2] https://github.com/a3f/openPOWERLINK_V2

Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

> *I think the notion of pipelining is the fundamental contribution of the [UNIX] system.*
>
> — Brian W. Kernighan, Bell Labs UNIX Werbung[3]

> *./extract-vmlinux /boot/vmlinuz-4.4.0-97-generic | size -*
> *size: '-': No such file*
>
> — GNU size (GNU Binutils)

Die GNU Binary File Deskriptor API wurde erweitert, um das Lesen der Standard-Eingabe zu ermöglichen. Die binutils objdump, objcopy, nm, size, dlltool und nlmconv wurden erweitert, um davon Gebrauch zu machen. Patch v3 folgt nach Ende der Klausurphase (DEC Alpha Tests schlagen z.Zt. noch fehl[4]).

BISHERIGE PROJEKTE

> *Die Nutzung der bereits vorhandenen Netzwerkkarte und ihrer Hardware-Zeitstempelung würde den Paketanalyseprozess erheblich optimieren, da die temporäre Beschaffung des SharkTaps vom Kollegen entfällt.*
>
> — Motivation

Vertraut machen mit Wireshark-Interna, insbesondere deren IPC mit dem externen dumpcap-Prozess. Erweiterung Wireshark und TShark, um Zeitstempelungskonfiguration an libpcap weiterzupropagieren.

[3] https://youtu.be/tc4ROCJYbm0?t=358
[4] https://sourceware.org/ml/binutils/2018-02/msg00452.html

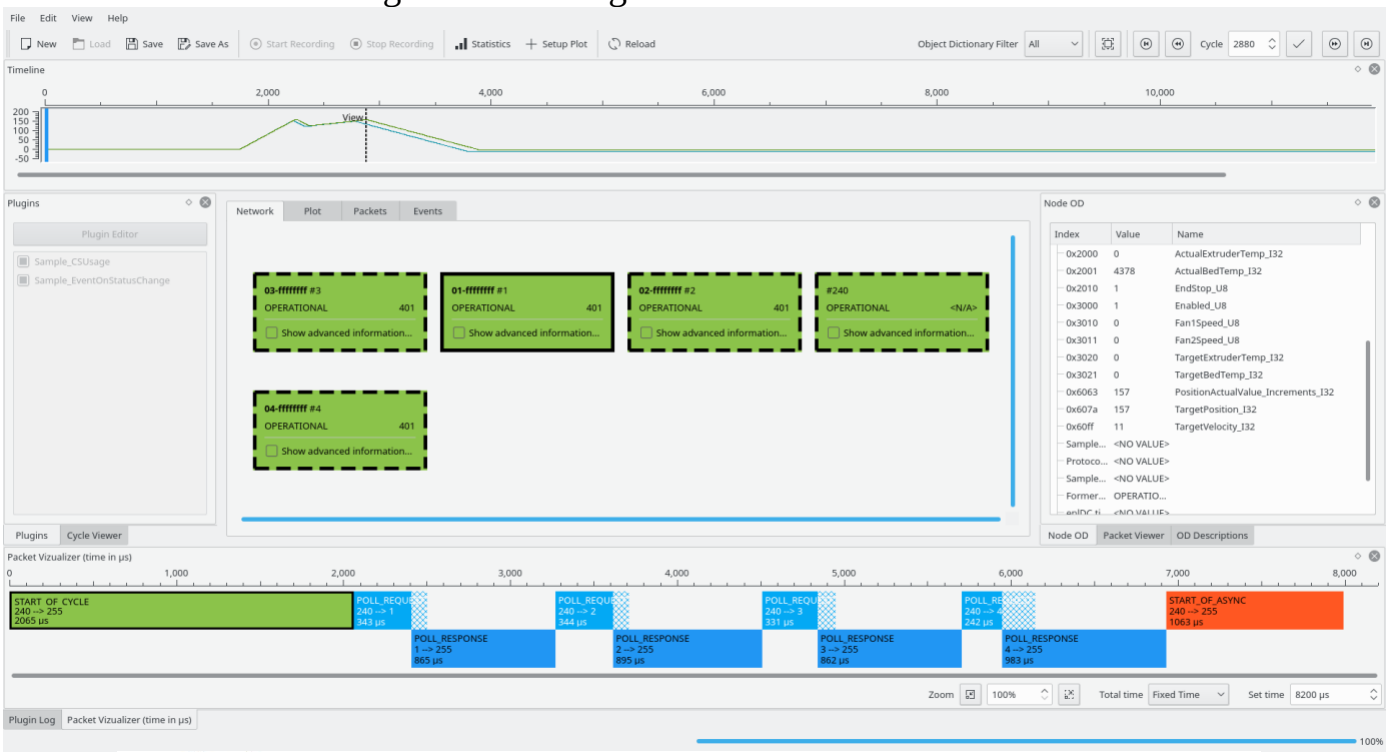
Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

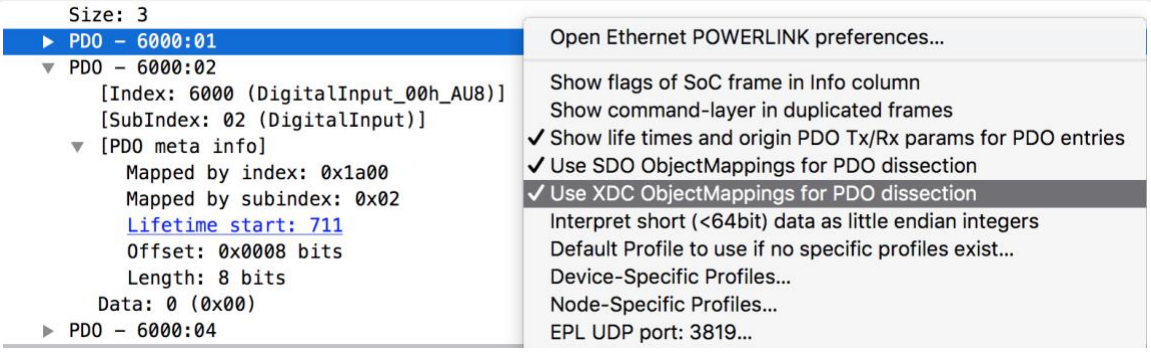
Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik

— "Praxis der Softwareentwicklung" Veranstaltung, KIT

> *Betreuer überzeugen in C programmieren zu dürfen.*
>
> — Motivation

Das Endprodukt, EPL-Viz [5], ermöglicht die graphische Visualisierung von Ethernet POWERLINK (EPL) Netzwerkverkehr.
Ich habe mich mit dem Einsatz von Wireshark als Unterbau für die Applikation beschäftigt. Dies beinhaltet Wrappen der (privaten) Wireshark-API, Lobbying um LibXML2 als Wireshark-Dependency hinzuzufügen, Implementierung von Parsern für die CANopen EDS- und EPL XDD-Formate und Erweiterung des EPL-Dissektors, dass er entsprechend dieser Gerätebeschreibungen und vorhergehenden Netzwerkverkehrs die PDUs seziert.





[5] https://github.com/epl-viz

Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

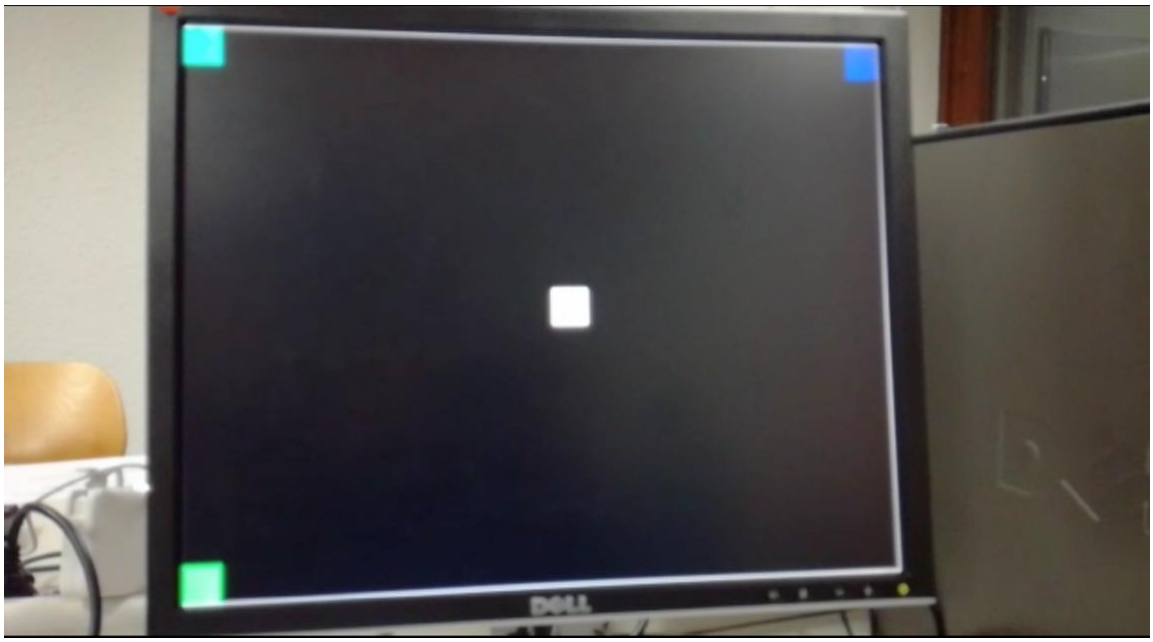
> *Warum sich einigen, welches Spiel wir auf dem FPGA implementieren sollen, wenn wir einen Prozessor entwerfen können, um dann beide Spiele herüber zu portieren?*
>
> — Motivation

”

Yeah, it runs DOOM[6]

— Tatsächliche Motivation

Im Rahmen des Praktikums, entstand in Zusammenarbeit mit einem Kommilitonen ein MIPS R3000 Softcore[7] (jedoch ohne MMU und Privilegienlevel).



Kein DOOM, aber erster Gehversuch Maschinencode auszuführen

Der Softcore kann rechnen (7 MIPS), speichern und LEDs, Knöpfe und 300 Byte VRAM per MMIO bedienen. Beläuft sich, mit Testbenches, auf rund 5000 Zeilen VHDL. Zum Portieren der eigentlichen Spiele war das Semester dann doch zu kurz.

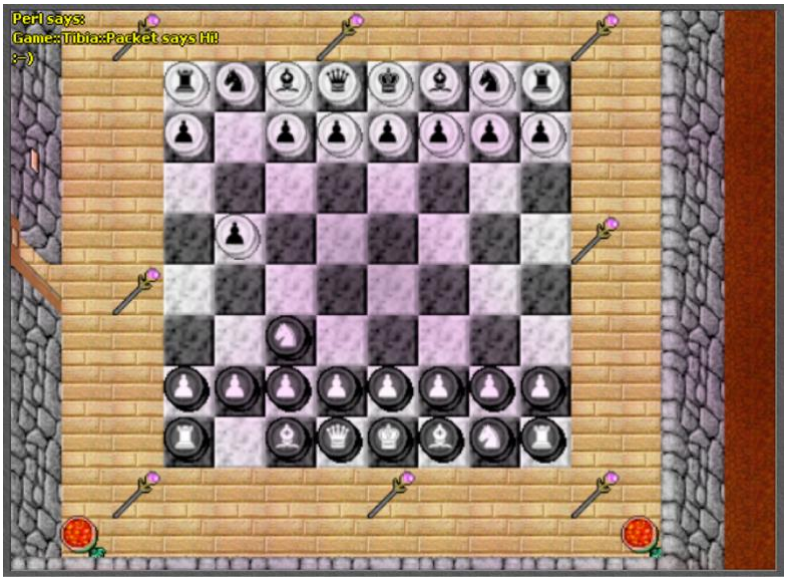
[6] http://itrunsdoom.tumblr.com/
[7] https://github.com/a3f/r3k.vhdl

Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

> *Good artists copy, great artists steal.*
>
> — Steve Jobs

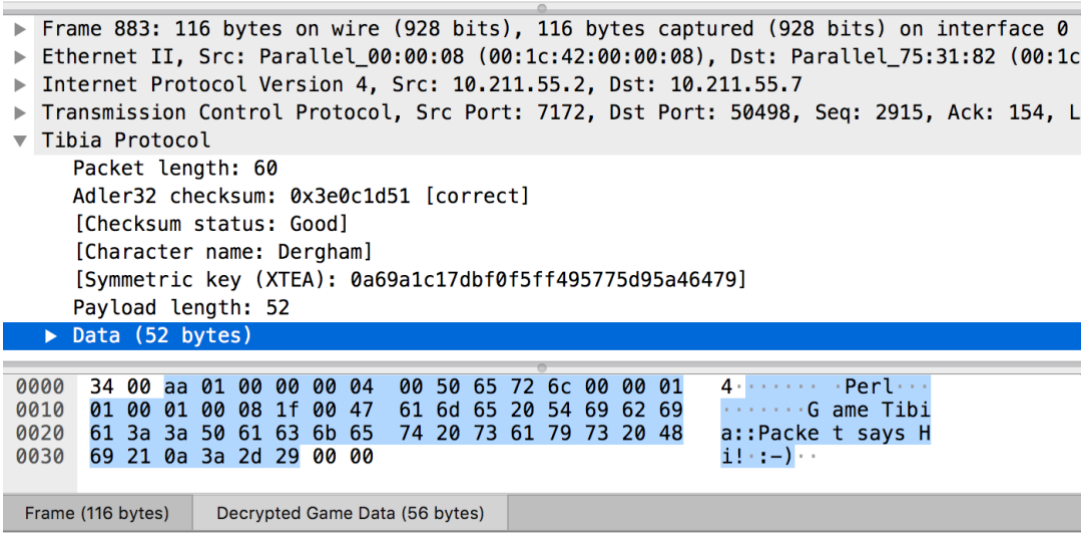
> *Some major cities have Game Rooms where players can find a friendly distraction from their normal routines and engage in a competition of skill, intellect or strategy by playing various Games.*
>
> — TibiaWiki, a wiki about the MMORP game Tibia

Die RSA-Routinen von Wireshark wurden refaktorisiert, um in anderen Dissektoren wiederverwendbar zu sein.
Das Netzwerkprotokoll des Spieles Tibia wurde reverse-engineert und bei Wireshark als Dissektor beigetragen. Ein Schachfeld wurde online präpariert, Map-Pakete wurden gedumpt und Perlskript wurde verfasst, welches zuerst das Paketdump an den Klienten versendet und dann die ItemMove und PlayerSay PDUs implementiert, um eine Schach-GUI zu bewerkstelligen:
das eigentliche Ziel des Projekts.



Das Projekt zeugt vom hohen Glauben des Bewerbers an die Rückwärtskompatibilität: der Wireshark-Dissektor unterstützt Protokoll-Versionen 7.0 bis 11.0[8], eine Zeitspanne von 15 Jahren, in schönem, portablem C89 Code.

Betroffene Projekte: Wireshark, Game::Tibia::Packet, Game::Tibia::Cam, Crypt::XTEA

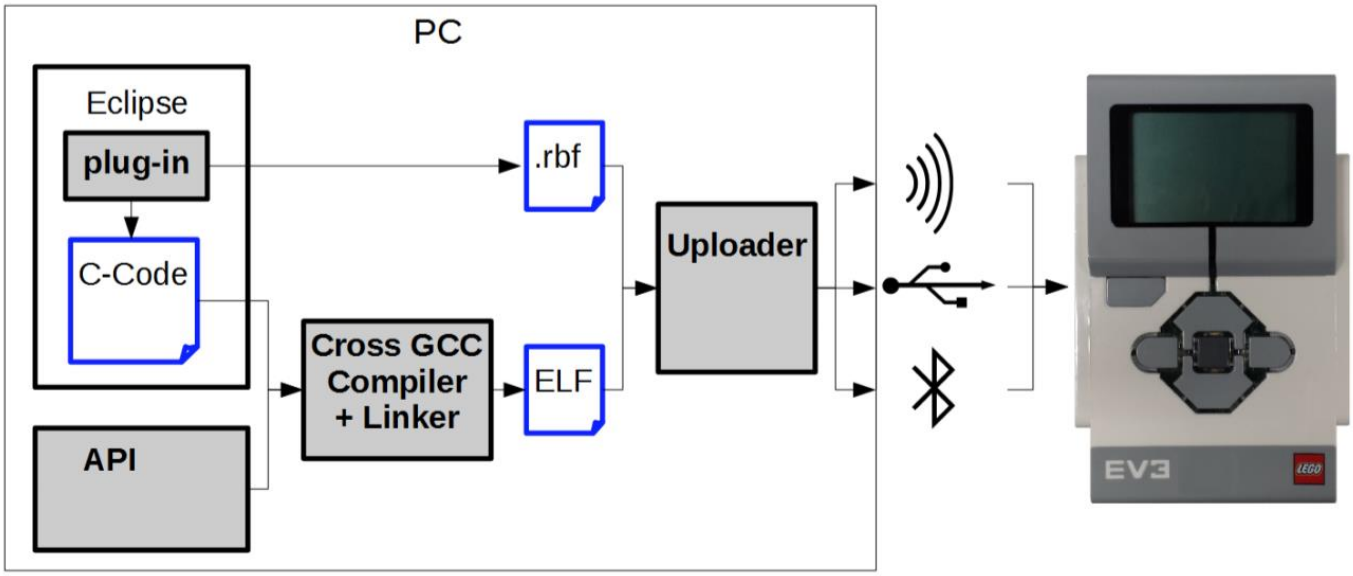


[8] Support für 7.70-7.72 momentan mangelhaft, aber Aufnahme von 7.70 wurde mir zugemailt und Support folgt voraussichtlich nach Ende der Klausurphase.

Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

> *Programming for a robot can give engineering students much more practical experience with technical problems and improve the learning motivation. This was the reason to include the EV3 at the university's computer science course in the studies for Mechatronics, which utilises ANSI C as the programming language.*
>
> — Bewerber et al., EDUCON 2016 Paper[9]

Erstellung c4ev3[10], ein cross-Plattform Development Kit für die Linux-basierten LEGO EV3 Roboter (Bluetooth, TCP und USB Uploader, Eclipse-Plugin, Toolchain, Unterstützung bei der Kontrollbibliothek) für die Nutzung im Lehrbetrieb an der Hochschule Aschaffenburg.
Neben den angehenden Elektrotechnikern und Mechatronikern, die seit 2015 zusammen mit c4ev3 ihre erste Informatik-Veranstaltung bestreiten dürfen, hat das Projekt aber auch freiwillige Nutzer[11].



(Das .rbf ist Bytecode für die LEGO VM und dient dem Bootstrapping)

[9] S. R. Perez, A. Fatoum and J. Abke, "Development of an Eclipse plug-in for using the LEGO Mindstorms EV3 in education," 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, 2016, pp. 631-636. http://ieeexplore.ieee.org/abstract/document/7474616/
[10] https://c4ev3.github.io/
[11] https://www.youtube.com/watch?v=jImqJCL3Tvs

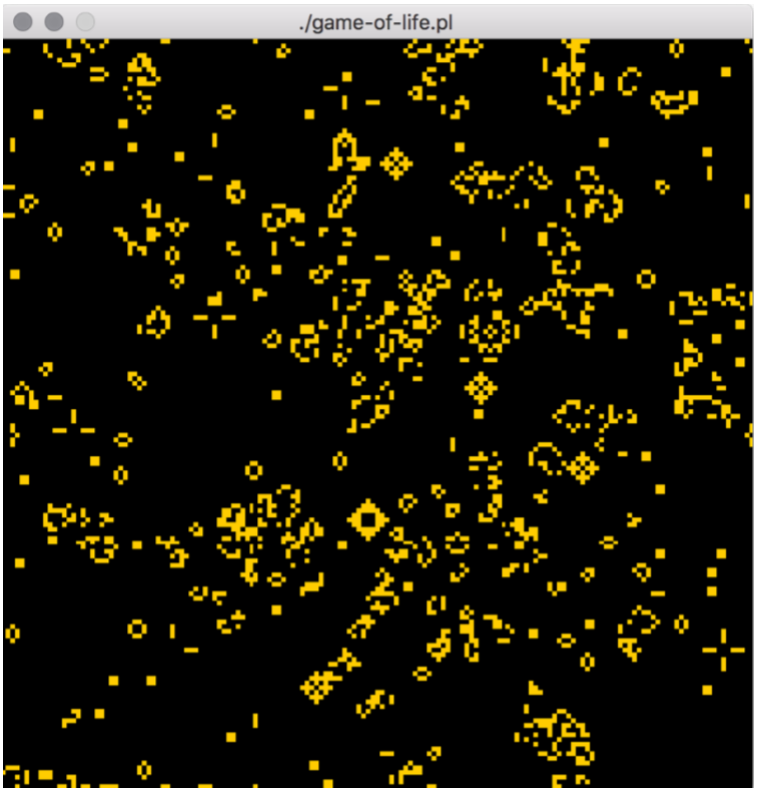
Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

```
life←{            ⍝ John Conway's "Game of Life".
    ↑1 ⍵∨.∧3 4=+/,¯1 0 1∘.⊖¯1 0 1∘.⌽⊂⍵
}
```

— DYALOG APL Einzeiler

> *Das will ich in Perl haben.*
>
> — Motivation

Angefangen mit Perl-Bindings[12] zur raylib-Videospiel-Bibliothek, um Conway's Game of Life zu visualisieren. Bei CPANtesers testen Hunderte von Maschinen diverser Konfigurationen Perl-Module, und ich schätze möglichst viele grüne Reports zu meinen Modulen. Daher kümmere ich mich[13] jetzt upstream um CMake, continuous integration, automatisches Deployment, Portierung und das Smoke-Testen von raylib.



(Perl-Reimplementierung des Einzeilers kommt - ohne Grafik und mit PDL - auf 158 Bytes)

[12] https://metacpan.org/pod/Graphics::Raylib

[13] https://github.com/raysan5/raylib/commits?author=a3f

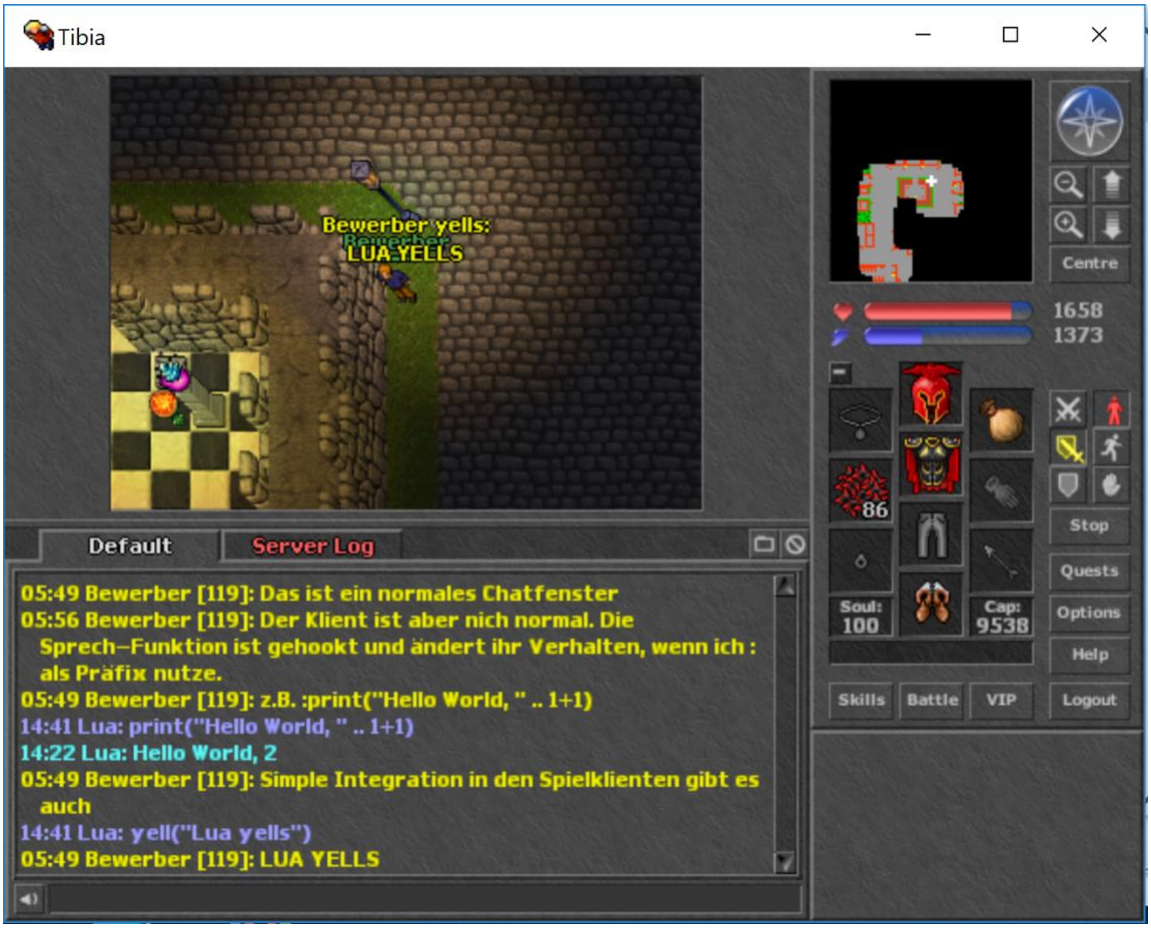
Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044

> *Lua is a fast language engine with small footprint that you can embed easily into your application. Lua has a simple and well documented API that allows strong integration with code written in other languages.*
>
> — https://www.lua.org/about.html “Why choose Lua”

”

Einen Lua-Interpreter in das Spiel hinein zu hacken, wäre sicher amüsant.

— Motivation



Chat-Subsystem des Spiel-Klienten wurde reverse-engineert. Ein Programm, welches ein Chatfenster zum Terminal umfunktioniert, wurde implementiert, als DLL mit dem Lua Interpreter gebündelt und in den Klienten injiziert.

Betroffene Projekte:
libvas[14]: Manipulation des virtuellen Adressraums über Prozessgrenzen hinweg auf Windows, Linux, macOS, BSD und Solaris.
Lade[15]: Laden von DLLs in den Adressbereich fremder Prozesse
ia32hook[16]: Simpler IA32 Funktionshooker, um Funktionsverläufe umzubiegen.

[14] https://github.com/a3f/libvas
[15] https://github.com/a3f/lade
[16] https://github.com/a3f/ia32hook

Ahmad Fatoum (ahmad@a3f.at), Friedenstr. 18, 76337 Waldbronn, Tel.: 01521 809 1044