Dynamische Programmaktualisierung vor 20 Jahren

Ausgewählte Kapitel der Systemsoftwaretechnik: Rekonfigurierbare Systemsoftware

Christoph Erhardt

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen-Nürnberg

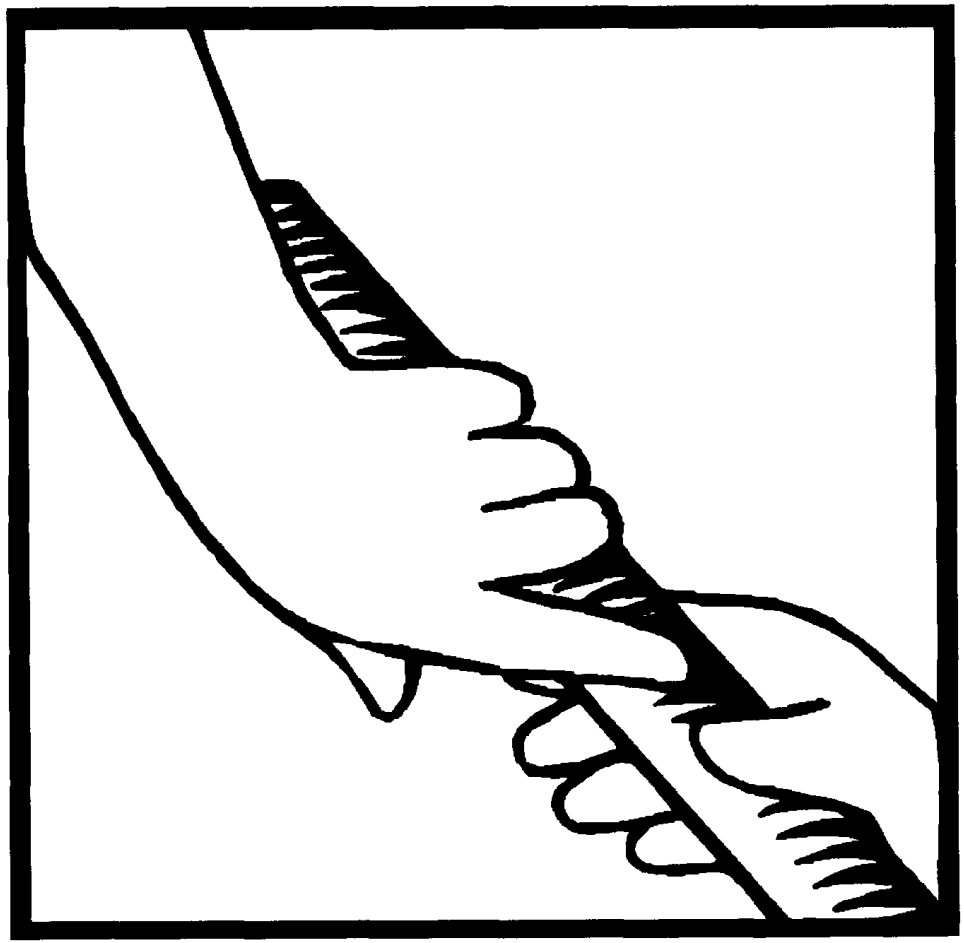
9. Mai 2014

https://www4.cs.fau.de/Lehre/SS14/MS_AKSS/

ON-THE-FLY PROGRAM MODIFICATION: SYSTEMS FOR DYNAMIC UPDATING

The rising cost of shutting down systems for maintenance and repair is forcing developers to look at ways to repair software as it runs. The authors briefly describe available updating systems and present a prototype.

MARK E. SEGAL
Bellcore
OPHIR FRIEDER
George Mason University



On-the-Fly Program Modification

Papier

- Journal-Papier
- Erschienen in *IEEE Software*, März 1993

→ Format und Umfang typisch für ein Journal-Papier

Autoren

- Mark E. Segal, Bellcore
- Ophir Frieder, George Mason University

→ Kooperation zwischen Industrie und Hochschule

Überblick

- Analyse existierender dynamischer Update-Verfahren
- Vorstellen eines neuen, eigenen Ansatzes: *PODUS*

Anno 1993 ...


intel®
pentium™
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992


intel®
pentium™
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992

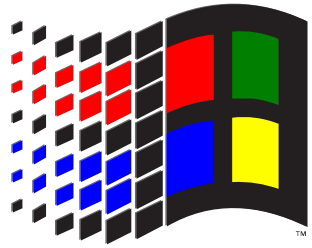

IBM PowerPC 601
© IBM 1992
PPC601FD-080-2
19425030KS

MICROSOFT
WINDOWSNT

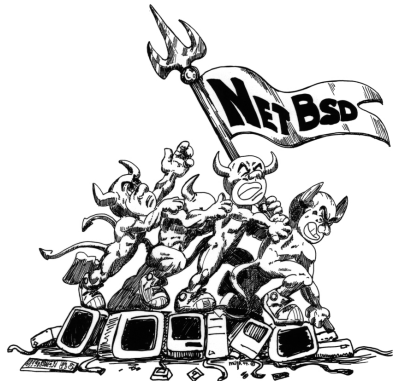

intel
pentium
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992


IBM PowerPC 601
© IBM 1992
PPC601FD-080-2
19425030KS


MICROSOFT
WINDOWSNT

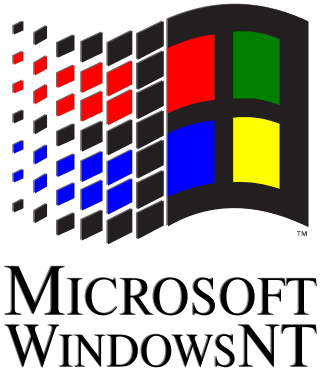


MICROSOFT
WINDOWSNT

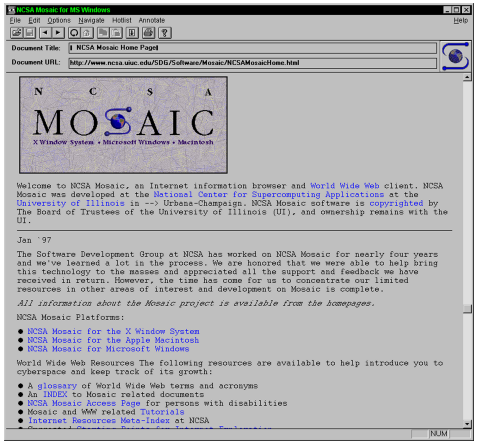



intel
pentium
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992


IBM PowerPC 601
© IBM 1992
PPC601FD-080-2
19425030KS

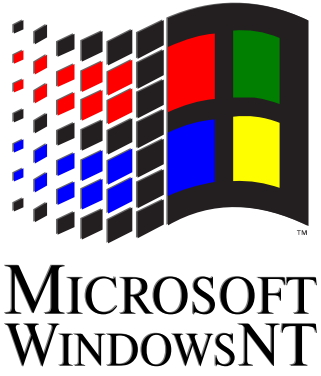

MICROSOFT
WINDOWSNT


NetBSD

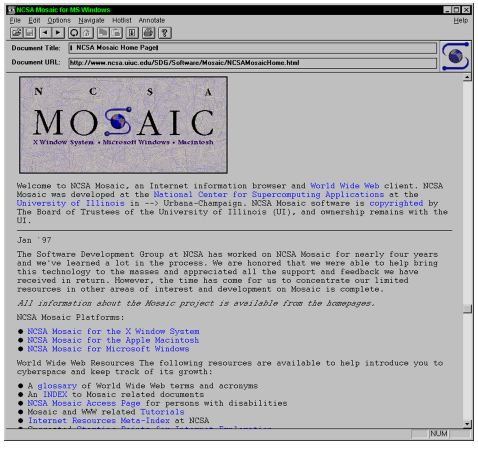

NCSA Mosaic for MS Windows
File Edit Options Navigate Hotlist Annotate Help
Document Title: NCSA Mosaic Home Page!
Document URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in --) Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.
Jan '97
The Software Development Group at NCSA has worked on NCSA Mosaic for nearly four years and we've learned a lot in the process. We are honored that we were able to help bring this technology to the masses and appreciated all the support and feedback we have received in return. However, the time has come for us to concentrate our limited resources in other areas of interest and development on Mosaic is complete.
All information about the Mosaic project is available from the homepages.
NCSA Mosaic Platforms:
● NCSA Mosaic for the X Window System
● NCSA Mosaic for the Apple Macintosh
● NCSA Mosaic for Microsoft Windows
World Wide Web Resources The following resources are available to help introduce you to cyberspace and keep track of its growth:
● A glossary of World Wide Web terms and acronyms
● An INDEX to Mosaic related documents
● NCSA Mosaic Access Page for persons with disabilities
● Mosaic and WWW related Tutorials
● Internet Resources Meta-Index at NCSA
NUM

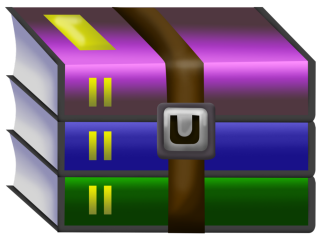

intel
pentium
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992


IBM PowerPC 601
© IBM 1992
PPC601FD-080-2
19425030KS


MICROSOFT
WINDOWSNT

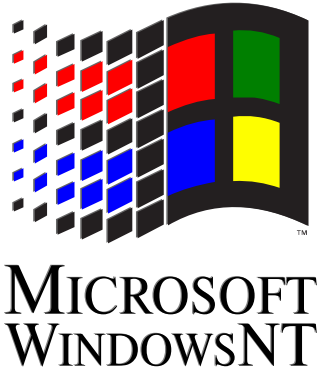

NetBSD


NCSA Mosaic for MS Windows
File Edit Options Navigate Hotlist Annotate Help
Document Title: NCSA Mosaic Home Page!
Document URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in --) Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.
Jan '97
The Software Development Group at NCSA has worked on NCSA Mosaic for nearly four years and we've learned a lot in the process. We are honored that we were able to help bring this technology to the masses and appreciated all the support and feedback we have received in return. However, the time has come for us to concentrate our limited resources in other areas of interest and development on Mosaic is complete.
All information about the Mosaic project is available from the homepages.
NCSA Mosaic Platforms:
NCSA Mosaic for the X Window System
NCSA Mosaic for the Apple Macintosh
NCSA Mosaic for Microsoft Windows
World Wide Web Resources The following resources are available to help introduce you to cyberspace and keep track of its growth:
A glossary of World Wide Web terms and acronyms
An INDEX to Mosaic related documents
NCSA Mosaic Access Page for persons with disabilities
Mosaic and WWW related Tutorials
Internet Resources Meta-Index at NCSA
NUM

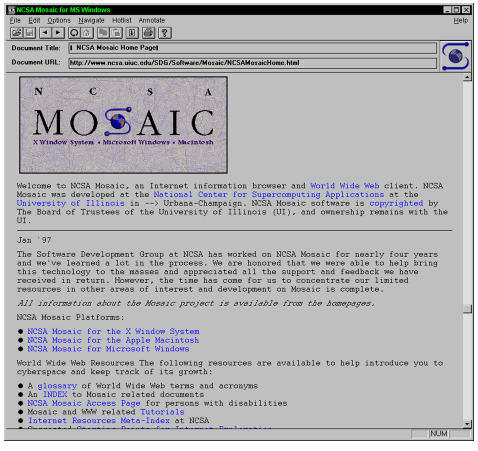


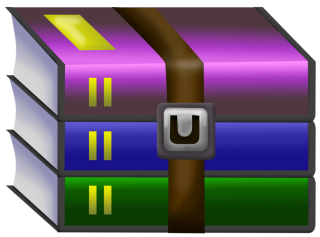

intel
pentium
PROCESSOR
A80501-60
L3342097
SX753
INTEL©1992


IBM PowerPC 601
© IBM 1992
PPC601FD-080-2
19425030KS


MICROSOFT
WINDOWSNT


NetBSD


NCSA Mosaic for MS Windows
File Edit Options Navigate Hotlist Annotate Help
Document Title: NCSA Mosaic Home Page!
Document URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in --) Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.
Jan '97
The Software Development Group at NCSA has worked on NCSA Mosaic for nearly four years and we've learned a lot in the process. We are honored that we were able to help bring this technology to the masses and appreciated all the support and feedback we have received in return. However, the time has come for us to concentrate our limited resources in other areas of interest and development on Mosaic is complete.
All information about the Mosaic project is available from the homepages.
NCSA Mosaic Platforms:
NCSA Mosaic for the X Window System
NCSA Mosaic for the Apple Macintosh
NCSA Mosaic for Microsoft Windows
World Wide Web Resources The following resources are available to help introduce you to cyberspace and keep track of its growth:
A glossary of World Wide Web terms and acronyms
An INDEX to Mosaic related documents
NCSA Mosaic Access Page for persons with disabilities
Mosaic and WWW related Tutorials
Internet Resources Meta-Index at NCSA
NUM




DOOM

Struktur des Artikels

1. Motivation
2. Anforderungen an DSU-Systeme
3. Kategorisierung existierender Lösungen
4. Eigener Prototyp: *PODUS*
5. Ausblick

Wissenschaftlicher Beitrag

- Umfassende Literaturanalyse, Kategorisierung und Bewertung
- Vorstellen eines verbesserten DSU-Systems auf UNIX-Basis

Motivation

- Software in zunehmendem Maße größer und komplexer
- Fehlerkorrekturen, Hinzufügen neuer Features notwendig
- Ausfallzeiten durch Neustart oft nicht tolerierbar:

Raumschiff im Orbit

- Auf keinen Fall das Lebenserhaltungssystem temporär abschalten!

Vermittlungsstelle zwischen Kommunikationsnetzen

- Angepeilte Verfügbarkeit $> 99{,}999\,\%$
- $<$ 2 Stunden Ausfallzeit in 40 Jahren!

Allgemeine Anforderungen an DSU-Systeme

- Beibehalten der Korrektheit
  - Keine Aktualisierung zum falschen Zeitpunkt
  - Keine unzumutbaren Leistungseinbußen
- Minimales menschliches Eingreifen
- Unterstützung für verschiedene Änderungsarten
  - Geänderte Implementierung
  - Geänderte Schnittstelle
  - Restrukturierung
- Gute Portabilität
  - Keine speziellen Hardware-Anforderungen
- Keine Einschränkungen hinsichtlich Anwendungsentwicklung
  - Programmiersprache
  - Programmierstil

Allgemeine Anforderungen an DSU-Systeme

- Beibehalten der Korrektheit
  - Keine Aktualisierung zum falschen Zeitpunkt
  - Keine unzumutbaren Leistungseinbußen
- Minimales menschliches Eingreifen
- Unterstützung für verschiedene Änderungsarten
  - Geänderte Implementierung
  - Geänderte Schnittstelle
  - Restrukturierung
- Gute Portabilität
  - Keine speziellen Hardware-Anforderungen
- Keine Einschränkungen hinsichtlich Anwendungsentwicklung
  - Programmiersprache
  - Programmierstil

Any program can be so poorly written that it cannot be dynamically updated.

Kategorien von DSU-Systemen

1. Hardware-basierte Lösungen
2. Ersetzen abstrakter Datentypen
3. Austausch von Servern in Client-Server-Umgebungen
4. Spezielle Nachrichtenaustausch-Systeme
5. Aktualisierung von prozeduralen Programmen

Hardware-basierte Lösungen

- *Hot-Spare*, Umschaltung auf aktualisiertes System
- Erhebliche Kosten
- Systeme müssen konsistent gehalten werden
  $\rightarrow$ Skaliert nicht
- (Geringer) Zustandsverlust?
- **Beispiel:** *Service Control Point* (Bellcore)
  - *In [the] worst-case scenario, the update takes approximately two weeks to plan and eight hours to complete.*

Ersetzen abstrakter Datentypen

- Aktualisierung der Implementierung bei unveränderter Schnittstelle
- Änderung des internen Formats erfordert Schreiben einer Konvertierungsfunktion
- **Beispiel:** *Dynamically Alterable System* (TU Berlin, 1978)
  - Granularität: Modul
  - Nutzt Hardware-Besonderheiten der PDP-11
    - Adressraumwechsel bei Sprung über Modulgrenzen hinweg
    - Verkettete Liste von Adressraumdeskriptoren
  - Aktualisierung = Manipulation der Deskriptorliste
  - Ineffizient und hochgradig unportabel

Austausch von Servern in Client-Server-Umgebungen

- Granularität: ganzer Dienst
- Aktualisierung der Implementierung bei unveränderter Schnittstelle
- **Beispiel:** *Michigan Terminal System* (University of Michigan, 1967–88)
  - Registrierung von Diensten am Dienst-Manager
  - Austausch jederzeit möglich
  - Laufende Anfragen werden mit alter Version fertig bearbeitet

Spezielle Nachrichtenaustausch-Systeme

- Aktualisierung verteilter Programme
- **Beispiel:** *Conic* (Imperial College London, 1989)
  - Module werden über Konfigurations-Manager zusammengestöpselt
  - Modulinterna beliebig änderbar
  - Aktualisierung eines Moduls nur bei Inaktivität – durch Umstöpseln im Konfigurations-Manager
  - Eigene Programmiersprache mit Laufzeitsystem
    $\rightarrow$ Akzeptanz schwierig

Aktualisierung von prozeduralen Programmen

- Granularität: Prozedur
- **Beispiel:** *Duplex Multi-Environment Real-Time Operating System* (Bell Labs, 1983)
  - Indirektion zwischen Funktionsaufruf und -implementierung
  - Keine Änderung von Schnittstellen
  - Kein Zustandstransfer
- **Beispiel:** *Dynamic Modification System* (University of Wisconsin–Madison, 1983)
  - Implementierung in Modula-ähnlichem Dialekt mit eigener Toolchain
    $\rightarrow$ Akzeptanz schwierig
  - Schnittstellenänderung möglich
  - Aktualisierung erfordert Handarbeit: `update A when X, Y, Z idle`
  - Overhead durch Locking-Protokoll zum Schutz aktiver Prozeduren

Zwischenbilanz

Alle Ansätze bisher nicht wirklich das Gelbe vom Ei...

Procedure-Oriented Dynamic Updating System

- Ansatz der Autoren
- Prototyp eines ersten „fortschrittlichen“ DSU-Systems

Anforderungen ans System

- UNIX-System (z. B. SunOS)
- Prozedurale Programmiersprache (z. B. C)
- Leicht modifizierter Compiler + Linker

Anforderungen an den Programmierer

- *Top-Down*-Programmierung: vernünftige Prozedurhierarchie
- Gekapselter Zugriff auf globale Daten

Features von PODUS

Ändern des internen Datenformats

- Handgeschriebene Abbildungsfunktion

Schnittstellenänderungen

- Handgeschriebene Wrapper-Funktion (*Interprocedure*)
- Konvertiert Argumente und Rückgabewerte vom alten ins neue Format

Mehrfädige Programme

- Nicht unterstützt

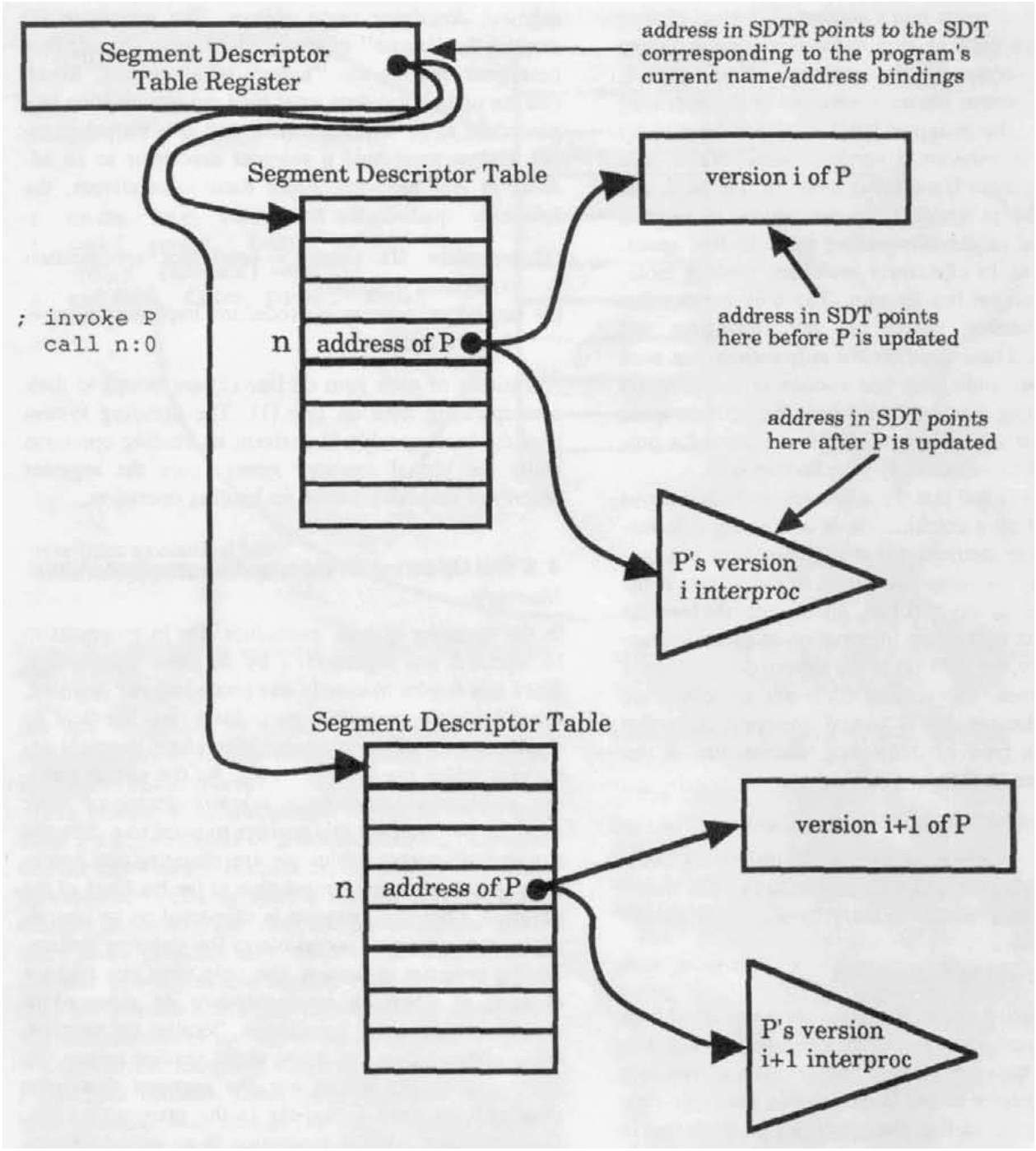
Implementierung

Einspielen von Aktualisierungen

- Verwaltungs-Code Teil des Binarys
- Ansprechbar über IP-Socket
- Einspielen von Aktualisierungen von außen per Update-Shell

Prozessstruktur

- Dünn besetzter Adressraum
- Strukturierte Adressen mit impliziter Codierung der Programmversion
- Bei entsprechender Hardware-Unterstützung: ein Segment pro Programmversion


Segment Descriptor Table Register
address in SDTR points to the SDT corrresponding to the program's current name/address bindings
Segment Descriptor Table
version i of P
address in SDT points here before P is updated
; invoke P
call n:0
n
address of P
address in SDT points here after P is updated
P's version i interproc
Segment Descriptor Table
version i+1 of P
n
address of P
P's version i+1 interproc

Ablauf einer Aktualisierung

- Laden des neuen Codes
- Schrittweises Ersetzen der geänderten Prozeduren
  - Voraussetzungen:
    1. Alte Prozedur befindet sich aktuell nicht in Ausführung
    2. Neue Prozedur ruft (transitiv) keine andere Prozedur auf, die aktuell in Ausführung ist
  - Scannen des Laufzeit-Stacks und des Aufrufgraphen
  - Neuscannen immer dann, wenn `return` dazu führt, dass Aufrufstack weniger Frames enthält als beim letzten erfolgreichen Ersetzen

$\rightarrow$ Blattprozeduren werden zuerst ersetzt, `main()` nie

Evaluierung

- *Unfortunately, we cannot compare PODUS's performance to that of other software-based updating systems because, to our knowledge, no performance data on these systems has been published.*
- *For one application – the updating of a hypothetical Internet packet router – we did a timing analysis. The analysis revealed that throughout the actual update, there was relatively little system degradation; the router continued processing at 90-percent efficiency.*

Zusammenfassung

- Praxisrelevantes DSU-System
  - Effizienz durch pfiffiges Ausnutzen von Segmentierung
  - Automatisches Bestimmen des jeweils richtigen Aktualizierungszeitpunkts
- *If indirection cannot be incorporated into a language or its underlying runtime system, dynamic updating cannot be done.*

Ausblick

- Bessere Werkzeugunterstützung
- Ausweitung auf andere Programmiersprachen und -paradigmen (z. B. funktional, objektorientiert)
- Mehrfädige Programme
- Unterstützung für Echtzeitsysteme

Diskussion

- Welche Relevanz hat PODUS im Jahr 2014?
- Welche Prinzipien des Papiers treffen heute noch zu?
- Welche Annahmen sind nicht mehr zeitgemäß?
- Warum hat sich DSU bis heute noch nicht im Mainstream durchgesetzt?