

startup	1
Europäisches Forum Für OS-9	2
EFFO Forum 20	11
OS-9 ftp-Archiv in der Schweiz	14
Öffentlich zugängliche OS-9 Software	16
Ein OS-9 Geräte-Treiber	18
Erkennen von Exceptions	25
A Software Management System	30
In eigener Sache	35
Vorschau	37

Herausgeber:
Marc Balmer
Hagentalerstrasse 12
CH – 4055 Basel

Telefon +41 61 281 19 41
Telefax +41 61 281 19 53
E-Mail os9-info@urz.unibas.ch

Redaktion:
Marc Balmer
Carsten Emde

startup

Vor Ihnen liegt die erste Ausgabe von **OS-9 International**, dem Magazin für OS-9 Anwender und Software-Entwickler. Von der Idee bis zur Realisierung dauerte es einige Zeit, doch hoffe ich, dass **OS-9 International** positiv aufgenommen wird und wir die für dieses Jahr geplanten Ausgaben herausgeben können.

OS-9 International soll der Verbreitung von Informationen rund um OS-9 dienen; es ist ausserdem Organ des EFFO, des Europäischen Forums für OS-9.

Beim Durchblättern wird Ihnen auffallen, dass **OS-9 International** mehrsprachig ist. Die Entscheidung, ob ein Artikel übersetzt wird oder ob er in der Originalsprache veröffentlicht werden soll, ist mir nicht leicht gefallen. Letztendlich bin ich aber zur Überzeugung gelangt, dass ein gut geschriebener Artikel in einer Fremdsprache immer noch besser ist als eine schlechte Übersetzung.

Ich stelle immer wieder fest, dass die OS-9 Gemeinde über ein grosses Potential verfügt, das aber relativ schlecht genutzt wird. Da die einzelnen OS-9 Anwender alle im "stillen Kämmerlein" sitzen und vor sich hin programmieren, bleibt viel Know-how der Allgemeinheit unzugänglich. Wenn sich jeder Programmierer zuerst seine Tools selber schreibt, wird das Rad unentwegt neu erfunden. Ein Erfahrungsaustausch dient deshalb allen OS-9 Anwendern.

Es gibt bereits einige regionale Benutzergruppen, die aber leider nicht viel Kontakt untereinander pflegen. Diese Gruppen und Vereine sind aufgerufen, sich zu melden und sich in **OS-9 International** vorzustellen.

OS-9 hat noch lange nicht die Beachtung gefunden, die es verdient. Daran können wir alle etwas ändern. Ganz besonders würde es mich freuen, wenn für die nächste Ausgabe von **OS-9 International** einige weitere Artikel von OS-9 Anwendern vorliegen würden.

OS-9 International möchte den wenigen Software-Häusern, die OS-9 Software entwickeln, unentgeltlich Gelegenheit geben, ihre Produkte bekannt zu machen. Aus diesem Grund werden Sie pro Ausgabe maximal einen produktbezogenen Artikel finden. Bezahlte Werbung hingegen ist deutlich gekennzeichnet.

OS-9 International ist nicht im Zeitschriftenhandel erhältlich. Der Bezug erfolgt ausschliesslich im Abonnement. Bitte sehen Sie dazu die Seite "In eigener Sache".

Marc Balmer

Europäisches Forum Für OS-9

Werner Stehling

OS-9 – was ist das? Wer diese Frage auf Anhieb beantworten kann, darf mit der weiteren Lektüre getrost aufhören. Für alle, die es nicht so ganz genau wissen, gestatten wir uns nur den leichten Hinweis: OS-9 ist ein Betriebssystem. Punkt. Also wirklich, im Gegensatz zu manchen anderen, die das auch von sich behaupten, ist es ganz bestimmt ein Betriebssystem.

Viel interessanter ist sowieso, was OS-9 nicht ist. Es ist z.B. denkbar ungeeignet, um mal schnell einen Brief zu schreiben. Erstens geht es nicht schnell und zweitens wird er nicht schön im Sinne der heutzutage verlangten inneren und äusseren Ästhetik, die sogar bei einem Schmierzettel mindestens Fotosatzreife voraussetzt. Ganz abgesehen von den Schwierigkeiten der dichterischen Schöpfung an sich – eine anständige Textverarbeitung unter OS-9 erschöpft sich auf dramatische Weise in metaphysischer Nichtexistenz. \TeX , werden Sie sagen. 'Ist das etwa nichts?' Na sicher doch, \TeX ist absolut das Grösste. Jedenfalls wenn man die paar 10 Megabyte auf der Platte betrachtet. Und ohne Previewer: wer zählt die Bäume, nennt die Namen, die ihr Leben für die 693 Probeausdrucke von Tante Bertas Geburtstagsgruss lassen mussten? Also lassen wir die Textverarbeitung mal beiseite und wenden uns etwas Erfreulicherem zu.

Nehmen wir die Tabellenkalkulation. Das braucht nun wirklich jeder: schnell ein paar Ausgaben zusammendividieren, eine Tabelle erstellen für den nächsten Quartalsbericht, die Auswirkung verschiedener Eingangsdaten feststellen, das alles ist eine der einfachsten Übungen für eine Tabellenkalkulation. Hurra, wie begeisternd, OS-9 ist nun wirklich das Betriebssystem, mit dem man um alle diese Probleme einen möglichst grossen Bogen machen sollte. Kalkulieren ja, meinetwegen auch Tabellen, aber beides zusammen?

Na ja, die Hauptdomäne von OS-9 ist natürlich bei den Datenbanken zu suchen, das wissen wir alle. Was spricht schon gegen 'suchen'? Suchen kann man immer, z.B. einen Record in der Datenbank. Oder ein Datenbankprogramm unter OS-9. Was haben die gemeinen Anwender da nicht schon alles gesucht, und wenn sie nicht gestorben sind, ...

CAD, klar, das hätte uns auch schon früher einfallen können, CAC, CAM und CIM, da liegen die Stärken von OS-9. Grafikdesign, Desktop-Publishing, alles

das sind Einsatzgebiete, die unter OS-9 perfekt und mit minimalem Aufwand angegangen werden können. Nahezu gleich Null ist der Aufwand, und wen stört schon, dass das Ergebnis auch nur unwesentlich von derselben verschieden ist?

Also vergessen wir OS-9. Wer braucht das schon. Benutzen wir einfach den vielgeschmähten Kompatiblen, der schafft spielend die Bearbeitung von 20 parallelen Echtzeitaufgaben, na gut, sagen wir 10, aber 5 sind es doch ganz sicher, wie bitte, 1.5 sagen Sie? Was funktioniert dabei nicht? Wieso kann man das nicht machen? Und wie gross dürfen die Programme sein? Ehrlich?

OS-9, das etwas andere Betriebssystem

Voller Stolz erlauben wir uns anzupreisen: OS-9 kann alles, weiss alles, macht alles - vielleicht nicht immer so perfekt wie andere, dafür aber manches, wovon andere nur träumen. Es gibt durchaus Textverarbeitung, Tabellenkalkulation, Datenbanken, wenn's denn unbedingt sein muss. Dafür gibt es ein paar Extras, die man sonst vergeblich sucht: Echtzeitverhalten, parallele Prozesse, minimale Systemanforderungen, modularen Aufbau, verschiebbare und von mehreren Benutzern gleichzeitig verwendbare Programme, ROM-Fähigkeit, alle denkbaren und undenkbaren Programmiersprachen und was der wunderbaren Eigenschaften mehr sind.

Warum, so schreckt der besorgte Leser auf, hat sich dieses Wundersystem nicht längst auf breiter Front durchgesetzt? Wieso findet sich im einschlägigen Fachzeitschriften-Blätterwald so gut wie nie auch nur ein Hinweis auf OS-9? Bitte kreuzen Sie den Vorschlag an, der Ihrer Meinung nach nicht die Ursache sein kann:

- ☐ Wenn alle diese Vorzüge von OS-9 öffentlich bekannt würden, wären alle anderen Systeme schlagartig vom Platz gefegt.
- ☐ OS-9 Plattformen sind viel zu teuer und die Lizenzgebühren sind viel zu hoch.
- ☐ Eine anständige Auflagenhöhe kommt nur zustande, wenn ein Betriebssystem möglichst viele Schwächen und Limitationen aufweist, über deren Behebung sich so herrliche Artikel schreiben lassen.
- ☐ OS-9 ist viel zu kompliziert, das kapiert der Leser nicht.
- ☐ Niemand kümmert sich ernsthaft darum, OS-9 in grossem Stil bekanntzumachen.

- Das Software-Angebot ist viel zu mager.

Womit wir jetzt ganz zwanglos die Überleitung zu EFFO gefunden hätten. Da es sich bei OS-9 offensichtlich um das beste Betriebssystem der Welt handelt, müssen wir es der Welt nur noch klar machen; aus völlig unbegreiflichen Gründen kommt sie nicht von alleine darauf.

EFFO, was ist das?

Die erste Bekanntschaft mit OS-9 endet normalerweise in einem Fiasko. Im Prospekt des Rechnerbauers liest sich alles so schön: Multiuser-Multitasking-Realtime-Betriebssystem, und dann sitzt man plötzlich und unerwartet vor einem Haufen Disketten und zwei dicken Handbüchern. Mit viel Glück läuft der Rechner auf Anhieb, aber wehe, ein Treiber stimmt nicht ganz oder der Descriptor muss an einer kleinen aber entscheidenden Stelle angepasst werden - wer von uns hat das nicht schon erlebt. Ohne jede Ahnung von dem neuen Betriebssystem ist an dieser Stelle erst einmal Feierabend. Zugegeben, nach der 20. Installation einer neuen Harddisk regt einen das nicht mehr auf, aber als Anfänger will der erste Superfrust erst einmal bewältigt werden.

Der zweite unerlaubte Tiefschlag ist gewissermassen schon vorprogrammiert: wenn die mitgelieferten (System-) Programme alle ausprobiert sind, wird die Preisliste nach ein wenig Software durchstöbert. Diese Suche ist normalerweise nach drei bis fünf Zeilen beendet, und der Sucher in der Wüste hat das dumpfe Gefühl, jetzt endgültig im Regen zu stehen: zum einen steht nichts Brauchbares drin und ausserdem kostet das auch noch ein Geld, ich meine richtiges Geld, nicht einfach nur Geld. Aha, denkt der Kandidat, OS-9 ist etwas für Spezialisten, die sich alle Programme selber schreiben, da gibt's halt nichts. In den Zeitschriften findet er sowieso nichts, und einen anderen OS-9 Anwender, den man fragen könnte, gibt es offenbar auch nicht. Das war's dann.

Nun ja, ganz so schlimm ist es auch wieder nicht, aber der Haken ist das gewusst wo. Längst nicht alle haben Verbindung mit einer Mailbox, die eventuell weiterhelfen kann. Die Wenigsten haben Zugriff auf das weltumspannende usenet mit seinen ftp Servern und riesigen OS-9 Konferenzen, und wer ist sich schon voll bewusst, dass fast das gesamte Potential an Unix-Programmen mit geringem Aufwand unter OS-9 läuft? Und wer sagt einem, dass diese ganze Pracht fast nichts kostet?

Genau an dieser Stelle setzt EFFO, das Europäische Forum Für OS-9, ein. Wir bieten einen Rahmen zur Kontaktaufnahme mit Gleichgesinnten, wir stellen

Lösungen für immer wiederkehrende oder auch ganz spezielle Probleme vor, wir verbreiten Informationen über OS-9, die nicht ohne weiteres zugänglich sind, und wir sorgen für die Verteilung vieler schöner Programme. Dazu ist uns beinahe jedes Mittel recht, wie Sie sehen, verschönern wir jetzt sogar gefällte, plattgewalzte, vermantschte und getrocknete Bäume mit Druckerschwärze. Allerdings hat eine Zeitschrift immer etwas von einer Einbahnstrasse; deshalb ist unser Hauptmedium die Diskette, die möglichst oft ihren Weg von Ihnen zu uns und wieder zurück finden sollte. (Da niemand genau weiss, wieviele Bäume für die Herstellung einer Diskette ihr Leben lassen müssen, haben wir hier ein völlig reines Gewissen.)

Die Teilnahme am Forum ...

Ca. 5 mal im Jahr organisieren wir einen Diskettenaustausch, bei dem alle Beiträge und Informationen von den Einsendern zusammengetragen und an die Teilnehmer zurückgeschickt werden. Dazu sendet jeder Interessierte zwei Disketten mit eigenen Beiträgen (oder zumindest formatiert) an den für das eigene Computersystem zuständigen EFFE-Redakteur ein (Adressen siehe unten) - und damit ist auch schon alles Wichtige gesagt. Eine noch simplere Möglichkeit ist uns bisher nicht eingefallen. Ältere Forumsausgaben sowie Disketten aus unserer PD-Sammlung sind ebenfalls erhältlich. Wie so etwas ungefähr aussieht, können Sie anhand des Inhaltsverzeichnisses von Forum 20 und aus unserer Liste mit PD-Disketten ansehen, die ebenfalls in dieser Ausgabe abgedruckt sind. Alle Beiträge können mittlerweile auch von einem ftp-Server in Basel abgerufen werden.

... und was es kostet

Um gleich der allgemeinen Verwirrung Einhalt zu gebieten, die beim Betrachten der finanziellen Aspekte jedweden zu übermannen droht, wollen wir hier einmal unsere Hochpreispolitik ausführlich erläutern. Getreu unserer Devise der absoluten Gewinnminimierung - was in letzter Konsequenz einen volkswirtschaftlichen Schwachsinn darstellt und eigentlich eine psychologische Unbedenklichkeitsbescheinigung erheischen sollte - kalkulieren wir äusserst knapp, aber brutal.

Was bekomme ich gratis?

Nichts natürlich. Jedenfalls nichts, was man anfassen kann. ABER, gesetzt den Fall, ich sende zwei formatierte Disketten ein, vielleicht sogar mit eigenem Gedankengut, einem kleinen Briefchen, einem Artikel oder einem Progrämmchen gefüllt, dann bekomme ich die Schlappscheiben sogar wieder. Und - das ist der Wahnsinnshammer - auf den Dingen ist auch noch mehr drauf als vorher. Der Bankkaufmann nennt so etwas 'Wertschöpfung', der Wissenschaftler 'Perpetuum Mobile', der Grüne wahrscheinlich 'Ausbeutung unterprivilegierter Umweltverbesserer' (bei den Grünen ist eine Definition immer etwas langatmiger und muss unbedingt den Begriff 'Umwelt' enthalten). Denn siehe, es steht geschrieben: 'EFFO lässt dort Bytes wachsen, wo vorher Wüste war und labt den Dürstenden aus nicht versiegendem Bitstrom'. Im Klartext: Die Teilnahme an der Forumsrunde ist absolut gratis und unverbindlich.

Was bekomme ich für 10 Schweizer Fränkchli?

Eine ganze Menge, vorausgesetzt, ich bin ein ordentliches EFFO-Mitglied. Für diesen Dumpingpreis liefern wir entweder eine alte Forumsausgabe oder eine Public Domain Diskette. Dabei handelt es sich immer um logische Ausgaben, d.h. z.B. i.a. zwei physikalische Forumsdisketten oder eine physikalische PD-Scheibe. Falls das ganze Paket auch in komprimierter Form nicht auf eine Kratzplatte passt, können es aber auch zwei oder drei lochzentrierte magnetpastenbeschichtete rundgestanzte schutzhüllte PVC-Flachprofile sein. Porto und Verpackung, sowie der Obulus für den Klapperteller selbst sind selbstverständlich eingeschlossen, ebenso natürlich sämtliche Mehr- oder weniger Wert- oder Warenumsatz-Steuern, die irgendwohin abzuführen uns von vornherein in den Sinn zu kommen versagt blieb.

Was bekomme ich für 20 Schweizer Fränkchli?

Einmal pro Jahr verschleudern wir dafür das allergrösste nur denkbare Beate-Uhse-Sonderangebot, nämlich eine Mitgliedschaft im EFFOclub. Damit verbunden sind a-tens Sitz und Stimme in der Vollversammlung, beta das Recht, ein öffentliches Amt bei EFFO zu bekleiden (trägt ein Amt eigentlich Hosen oder Röcke?) und drit-tens profitiert man von unseren zehnfränkigen Schleuderpreisen.

Alle diejenigen, die aus politischer, weltanschaulicher oder religiöser Überzeugung eine Mitgliedschaft ablehnen, erhalten für 20 Franken jeweils eine alte Forumsausgabe oder eine PD-Diskette, logischerweise wieder logische Ausgaben. Obwohl die Teilnahme am Forum kostenlos und unverbindlich ist, freuen wir uns ueber jeden, der zusaetzlich Vereinsmitglied bei EFFO wird. Die Mitgliederbeitraege werden verwendet, um die im Zusammenhang mit dem Forum anfallenden Kosten zu decken.

Voraussichtlich auf Forum 397 werden wir ein Programm veröffentlichen, das in jedem einzelnen Fall vorausbestimmen kann, ob sich eine Mitgliedschaft lohnt. Eine Arbeitsgruppe unserer Redakteure hatte bereits die erste Sitzung, und es steht auch schon die Sprache für das Programm fest: BASIC++. Es wurde ein Unterausschuss eingesetzt, um den notwendigen Interpreter zu erstellen, wahrscheinlich in COBOL. Mit an Sicherheit grenzender Wahrscheinlichkeit wird das fertige Programm einen PC AT SX mit VGA voraussetzen, der unter Windows 27 einen Atari mit mgr emuliert. Einfacher gehts halt wirklich nicht. Und so etwas hat eben auch jeder, oder wenigstens zu haben, denkt sich zumindest die Industrie, die von sich behauptet, einen Standard zu vertreten.

Mit einer gewissen Penetranz werde ich dabei immer an den Siegeszug des guten alten VW Käfers erinnert. Das Gefährt war hoffnungslos untermotorisiert, verfügte über die Fahrleistung einer Schildkröte, hatte nicht den geringsten (Kofferraum-) Speicher, den Bedienungskomfort eines Waschbretts und die Strassenlage einer Wassermelone. Aber das Ding hat sich durchgesetzt. Zugegeben, den Witz habe ich auch nie begriffen.

Was bekomme ich für 60 Schweizer Fränkchli?

Gesetzt den Fall, es rotten sich mehrere Gleichgesinnte zusammen und bilden eine nicht notwendigerweise terroristische Vereinigung, sondern z.B. eine Firma oder einen lockeren Haufen von Saufkumpanen, dann können sie eine Kollektivmitgliedschaft bei EFFO erwerben. Damit verbunden ist ein einzelner Sitz und Stimme in der Vollversammlung, und es nimmt auch nur ein Einsender an den Forumsrunden teil. Der Sinn ist wohl einigermassen klar: Arbeitsteilung. Ausserdem soll der finanzielle Anreiz den Herdentrieb von ansonsten einzelkämpferisch angehauchten OS-9-Frohnaturen fördern: eine gelegentliche direkte Kontaktaufnahme mit Gleichdenkenden zum Zwecke des Informations- und Gedankenaustauschs kann durchaus befruchtend wirken. Das geliebte Eheweib ist häufig auch ganz froh, wenn der Göttergatte ab und zu das Feld räumt. Und gibt es etwa

eine schlagkräftigere Methode, den Haushaltsvorstand von eventuell aufkeimenden Begleitbestrebungen abzuschrecken, als ein Computerklöbb?

Wer bekommt wieviel Rabatt?

Vergleicht man einmal unsere doch ziemlich überhöhten Preise mit den gängigen Klöbbbeiträgen, die den armen fehlgeleiteten Besitzer eines Industrie-Standards in seiner natürlichen Verzweiflung voll erwischen, dann sieht das Verhältnis sehr ungünstig aus. (Im tiefsten Inneren unseres Herzens wissen wir selbstverständlich, dass unser missionarischer Einsatz bei den Kompatiblen und solchen, die es sein wollen, um Größenordnungen wichtiger wäre, als bei OS-9, aber wir schaffen es nicht, genügend Mitleid zu empfinden, wir schaffen es einfach nicht.) Jedenfalls sahen wir uns deshalb genötigt, eine ziemlich ausgeschlafene Rabattstaffel einzuführen.

1. Schüler, Studenten, Rentner und Invalide erhalten einen generellen Nachlass von 0%. Das berücksichtigt unter anderem auch die Tatsache, dass die genannte Personengruppe heutzutage über Haushaltsmittel verfügt, die einem gestandenen Familienvater das schiere Salzwasser in die Augen treibt.
2. Alle, die einen eigenen Beitrag in irgend einer Form zum Forum einsenden, erhalten einen Rabatt von 100% auf die laufende Forumsrunde und von 0% auf ältere Ausgaben und PD-Disketten. Dahinter steht die Idee, dass das Forum nur mit Hilfe von eingesandten Beiträgen funktioniert. Wenn wenig hereinkommt, dauert eine Runde eben länger; wenn nichts hereinkommt, ist die Idee gestorben und wir ersparen uns ziemlich viel Arbeit. Wenn sich aber möglichst viele an den Gedanken gewöhnen, dass ein eigener Beitrag allen anderen und letztlich auch ihm selbst nützen kann, dann sehen wir rosigen Zeiten entgegen.
3. Diejenigen, die nicht unter die vorhergehenden Gruppen fallen, erhalten einen pauschalen Rabatt von 0%. Damit möchten wir unserer Freude Ausdruck verleihen, dass Sie EFFO mit Wohlwollen gegenüber stehen und unsere Ziele tatkräftig unterstützen.
4. Die Rabattstaffel für ausgewählte Teile, sagen wir mal Forum 7, Disk 2, Tracks 13 bis 27, oder auch ein Mengenrabatt sind in ihrer Konzeption und in den Ausführungsbestimmungen noch komplizierter. Also fragen Sie besser gar nicht erst nach.
5. Einen Trick wollen wir aber noch verraten: wenn Sie Ihren Redakteur ganz lieb bitten, ob er nicht vielleicht einfach noch mal so schnell diese oder jene

Diskette beilegen könnte, dann wird er es möglicherweise sogar tun. Redakteure sind auch nur Menschen, und bekanntlich ist das Fleisch willig, aber der Geist ... ich sage nur: der Geist - wenn ich mir vorstelle, wie der ist ...

Sie können leicht ausrechnen, dass der EFFO-Jahresbeitrag ungefähr dazu ausreicht, das Rückporto für ca. 6 Forumsrunden zu bezahlen. Also heisst es überall zu sparen, denn wenn EFFO erst einmal pleite ist, dann ist EFFO pleite und dann gute Nacht, Marie. Allerdings sind wir durchaus bereit, bei grösseren Projekten einen finanziellen Beitrag zu leisten. Wenn z.B. jemand eine vollständige T_XImplementation auf 50 Disketten beisteuern möchte, dann werden wir sicher eine befriedigende Lösung für alle Beteiligten finden.

Das ist also in aller Kürze unsere Preisliste. Klar, übersichtlich und von geradezu dramatischer Schlichtheit. Übrigens: alle ordentlichen EFFO-Mitglieder haben natürlich das Recht, an der nächsten Generalversammlung mündlich oder schriftlich ihre eigenen Vorstellungen darzulegen.

Ressortleiter des EFFO

Hier sind die verantwortlichen Ansprechpartner für alle möglichen Arten von Problemen. Wir beantworten fast jede Frage, oder kennen zumindest jemanden, der jemanden kennt, der jemanden kennt,

ATARI, Macintosh

Philip Maechler, Grundwiesstrasse 15, CH-8700 Küsnacht/ZH
+41 1 910 41 85

c't68k

Guido Hollenstein, Obere Weidstrasse 8, CH-6343 Rotkreuz
G:+41 42 44 74 06, P:+41 42 64 25 59

GEPARD

Reto Peter, Wallisellenstrasse 483, CH-8050 Zürich
G:+41 1 236 66 75, P:+41 1 322 74 52

VME, GMX micro 20

Lukas Zeller, Postfach 1, CH-8024 Zürich
G:+41 56 83 33 77, P:+41 1 261'35'89

mc68000

Thomas G. Schiele, Winterthurerstrasse 297, CH-8057 Zürich
G:+41 1 236 47 61, P:+41 1 311 22 35

NDR-Klein Computer

Heinz Pauli, Gerbiweg 15, CH-8910 Affoltern a.A.
+41 1 761 10 22

PD Organisator

Stephan Paschedag, Bauernhalde 5, CH-8708 Männedorf
G:+41 56 83 33 77, P:+41 1 920 32 77

Redaktionsleitung

Werner Stehling, Seilerwis 3, CH-8606 Greifensee
G:+41 1 256 42 21, P:+41 1 940 05 00

Mitgliederverwaltung

EFFO, Postfach, CH-8606 Greifensee

Kontaktadresse Deutschland

Computer Verein (CoVe), Otto E. Tutzauer, Willibrachtstrasse 13,
D-W-6000 Frankfurt/M 50, +49 69 52 84 95

Postscheckkonto: EFFO 80-48'254-1 Zürich, Schweiz

Werner Stehling ist Physiker. Er beschäftigt sich beruflich an der ETH Zürich mit technischen Entwicklungen im Bereich der Radioastronomie. Werner Stehling ist seit vier Jahren Präsident des Europäischen Forum Für OS-9. Sie können ihn postalisch unter der folgenden Adresse erreichen: Werner Stehling, Seilerwis 3, CH-8606 Greifensee.

EFFO Forum 20

Inhaltsverzeichnis Disk 1

lies_mich	Informationen über das EFFO (deutsch)
read_me	Informationen über das EFFO (englisch)
contents	Dieses Inhaltsverzeichnis
deadline_??_??_??	Der Stichtag für das nächste Forum (siehe lies_mich)
highlights	Neuigkeiten
info_empty_form	WICHTIG FUER ALLE FORUMS-TEILNEHMER: Diese Datei ist eine Vorlage für das neue Software-Beschreibungs-Format, das wir für alle Software, die von EFFO verwaltet wird, einführen möchten. Bitte diese Datei anschauen und wenn möglich, Informationen zu zukünftigen Forumsbeiträgen in diesem Format liefern! Vielen Dank!

EFFO

statuten	Die aktuellen Statuten
anmeldung	Antrag für Mitgliedschaft
registration	Antrag für Mitgliedschaft (engl.)
termine_effo	Veranstaltungskalender
spielregeln	Regeln für die Teilnahme am Forum
who_is_who	Wer ist wer bei EFFO und im Forum
f_disks	Inhaltsverzeichnis früherer Forumsdisketten
pd_disks	Verzeichnis der EFFO PD Disketten
preise	Was mich EFFO kostet
mitglieder_91	Freigegebene Adressen der EFFO-Mitglieder
protokoll_effo_gv_1992	Protokoll der letzten EFFO-GV

INFO

opus_bbs	Anleitung der OS-9 EFFO OPUS-Mailbox
os9_bbs	Anleitung des OS-9 BBS Konferenz-Systems, der Software-Datenbank und der Mailbox
soft_list	Liste aller in der Datenbank des OS-9 BBS
soft_list_xref	eingetragenen Software für OS-9.

cern.91.12	CERN
cern.92.02	Neuigkeiten ueber OS-9, Fehler Reports (engl.)
CoVeLeistungen	Beschreibung der CoVe Leistungen
CoVe.SwLib	SW-Liste von CoVe (und teils EFFO)
cove_news_2	Alte CoVe-News für alle, die sie verpasst haben.
kurs.programm.OSK_92	Beschreibung des Kurses, den Cove im Mai 92 durchführt.
readme.CoVe	Einige allgemeine Informationen über CoVe
report.CoVe_91Q3	Quartalsberichte für das 3. und 4. Quartal 91
satzung.CoVe	Die aktuellen Statuten von CoVe
tips.CoVe.tnt	Neueste Version der Tips und Tricks von CoVe

LETTERS

brief.*	Briefe der aktiven Teilnehmer am Forum
---------	--

SOFTWARE/SPOOLER

pr_queue.c	Task, die in die Queue schreibt. Von print gestartet.
pr_spooler.c	Task, die von der Queue liest und auf den Drucker schreibt.
print.c	Programm zum Ausdrucken mittels der Queue. Der Druckvorgang ist Event gesteuert.
printer.h	Definition der Kontroll-Codes. Für Epson, NEC, Star.

SOFTWARE/PIPE

wise_crack*	wise_crack füllt eine bezeichnete Pipe mit sehr
_note_pipe	intelligenten Logos, ask holt die erste Zeile dieser
ask*	Pipe.

SOFTWARE/OS9LIB

MAN/*.prf	Manual Seiten der OS9lib in proff Format
DEFS/*.h	Header Dateien der OS9lib

Inhaltsverzeichnis Disk 2

SOFTWARE/OCOMPRESS

SOFTWARE/PROCCOMM

AttFils* Inter-Prozess-Kommunikation mit Beispielen, Brief

HARDWARE/CT68000/GIO_TRAPHANDLER

files.ausdrucken	Anleitung zum Ausdrucken der Textdateien.
giocharset.a	Zeichensätze für Block, Futur, Latein und Prestige. Diese Sätze werden auch von der Emulation von Forum 19 verwendet.
giodemo.a	Demo Programm, welches den Trap Handler verwendet.
giographic.a	GIO Traphandler
giographic.*	Definitionen
giographic.txt	Beschreibung des GIO Trap Handler
giolib.a	C Interface
giolib.h	Beschreibung der C Schnittstelle
giotest.c	Testprogramm der C Schnittstelle
makefile	Makefile des GIO Trap Handler

HARDWARE/GEPARD/FAT_FONT

fat_fontset	Ein fetter Zeichensatz für Vorträge
conv2effo	Ersetzt Umlaute durch Doppelbuchstaben
ptop, ptob	Für das Erzeugen und Zusammenstellen von Textdateien

HARDWARE/GEPARD/DRIVER

* Druckertreiber für Epson LQ 850 und Fat_Font

HINWEIS: Alle “*.hlp”, “*.hlib” und “*.ndx” -Files sind Help-Files fuer das neue hierarchische HELP-Utility (siehe PROGRAMME/C/HELP auf Forum.10). Sie sollten unverändert ins Directory /dd/SYS/HELP kopiert werden. Auch Manual-Pages “*.man” sowie die alten (original-Microware) Helpfiles (ohne Extension) gehören dorthin, damit HELP sie finden kann.

OS-9 ftp-Archiv in der Schweiz

Gute und preiswerte OS-9 Software konnte bisher praktisch nur auf Disketten bezogen werden. EFFE und auch TOP in München stellen auf Disketten Software von hoher Qualität gegen eine geringe oder gar keine Bearbeitungsgebühr zur Verfügung.

Für viele OS-9 Anwender ist jedoch ein Online Zugriff auf OS-9 Software angenehmer. Vor einiger Zeit hat denn auch ein amerikanischer Sysop auf einem Rechner am Internet ein Software-Archiv eingerichtet, das OS-9 Software weltweit zur Verfügung stellt¹. Da der Rechner in den USA steht, ist der Filetransfer nach Europa sehr teuer. Auch wenn die meisten Benutzer die Filetransfers nicht selber bezahlen müssen, entstehen für den Systembetreiber hohe Kosten.

Um OS-9 Software über Netzwerk auch in Europa zu geringeren Kosten verfügbar zu machen, wurde am Institut für Informatik der Universität Basel ein OS-9 Software Archiv eingerichtet, das öffentlich über ftp (file transfer protocol) zugänglich ist.

Das Archiv in Basel gleicht vom Aufbau und Inhalt her dem bereits genannten Archiv in Wisconsin.

Um das OS-9 Archiv benutzen zu können, müssen Sie Zugang zum Internet haben. Wenn Sie einen Account auf einem Uni-Rechner haben, steht Ihnen ftp im Allgemeinen zur Verfügung; fragen Sie im Zweifelsfall Ihren Sysop.

Das OS-9 Archiv ist unter dem symbolischen Namen *lucy.ifi.unibas.ch* oder der numerischen Adresse 131.152.81.1 erreichbar. Wie bei vielen ftp-Servern üblich, ist der Benutzername **anonymous**, die eigene Mail-Adresse dient als Passwort:

```
$ ftp lucy.ifi.unibas.ch ...
Name (lucy.ifi.unibas.ch:muster): anonymous
Password: muster@urz.unibas.ch
FTP> cd osk
FTP> help
```

¹hermit.cs.wisc.edu, IP Adresse 128.105.8.24

Das Passwort wird nicht angezeigt, im obigen Beispiel dient es nur zur Verdeutlichung.

Sie finden auf dem Server die Datei **1s-1R**, die den aktuellen Inhalt des Archivs enthält. Bitte durchsuchen Sie das Archiv nicht “von Hand”, die Datei wird laufend aktualisiert.

Für Uploads steht Ihnen das Verzeichnis **incoming** zur Verfügung. Dateien, die Sie in **incoming** ablegen, werden vom Sysop gesichtet und dann an den richtigen Ort kopiert. Um diese Arbeit zu erleichtern, sollte zu jeder Datei ein kurzer Kommentar per E-Mail an **os9-info@urz.unibas.ch** geschickt werden.

Öffentlich zugängliche OS-9 Software

Damit Sie eine OS-9 "Quelle" in Ihrer Nähe finden können, unterhält OS-9 International eine Liste von öffentlich zugänglichen Computersystemen, die OS-9 Software anbieten.

Die Liste gliedert sich in zwei Teile, Systeme mit Anonymous FTP Zugang und Systeme mit Modem-Zugang (Mailboxen, BBS).

Die Liste ist nicht vollständig. Wir bitten Sie daher, uns alle Ihnen bekannten Mailboxen oder FTP-Archive mitzuteilen, die OS-9 Software anbieten. Sie erreichen uns am schnellsten via E-Mail an **os9-info@urz.unibas.ch**.

Anonymous FTP

<i>Land</i>	FTP Adresse
<i>Deutschland</i>	reseq.regent.e-technik.tu-muenchen.de tupac-amaru.informatik.rwth-aachen.de ifi.informatik.uni-stuttgart.de athene.uni-paderborn.de sun0.urz.uni-heidelberg.de
<i>Finnland</i>	nic.funet.fi mcsun.eu.net uunet.uu.net
<i>Australien</i>	brologa.cc.uq.oz.au kirk.bu.oz.au
<i>Japan</i>	srawgw.sra.co.jp toklab.ics.osaka-u.ac.jp oskgate0.mei.co.jp
<i>Frankreich</i>	irisa.irisa.fr
<i>Schweden</i>	sunic.sunet.se
<i>Schweiz</i>	lucy.ifi.unibas.ch
<i>USA</i>	hermit.cs.wisc.edu wuarchive.wustl.edu

Mailboxen und BBS

<i>Land</i>	Mailbox	Modem-Nummer	Bps.
<i>Schweiz</i>	Gepard's Oracle	01 363 70 37	9600
	WOPR	061 281 55 52	11400
	(Username: os9)		
	EFFO BBS (über Kometh erreichbar)		
<i>USA</i>	The Penetrator BBS York	(717) 741 5078	2400
	Color Galaxy Milky Way	(415) 883 0696	2400
	(OS-9/6809 Network, 12 Nodes)		
	ACS, Inc. BBS (FIDO 133/510)	(404) 636 2991	???

Ein OS-9 Geräte-Treiber

Marc Bahmer

Das Betriebssystem OS-9 erfreut sich dank seiner Modularität in der Industrie vor allem als Steuerbetriebssystem grosser Beliebtheit.

Insbesondere die Möglichkeit, neue Hardware durch Treiber ins Betriebssystem² zu integrieren, trägt in hohem Masse zur Flexibilität und Beliebtheit von OS-9 bei.

Die Treiberprogrammierung, obwohl theoretisch nicht sonderlich schwierig, bietet in der Praxis doch meist einige Schwierigkeiten. Daher werden denn auch oft Peripheriebausteine direkt adressiert, statt einen OS-9 konformen Treiber zu schreiben.

Der grundsätzliche Aufbau eines OS-9 Treibers soll im folgenden beschrieben werden. Der hier vorgestellte Treiber erfüllt allerdings keine besonders wichtige Funktion, dient er doch lediglich dazu, die Statusanzeige (ein Digit Hex-Display) eines VMEbus-Computers (E5, ELTEC, Mainz, Deutschland) anzusteuern.

OS-9 kennt verschiedene Treiber-Arten

Nicht alle Peripheriegeräte lassen sich in gleicher Weise ansprechen. Ein Bildschirm-Terminal empfängt zum Beispiel ein Zeichen nach dem anderen über eine serielle Leitung. Es ist also ein "Zeichen-für-Zeichen"-Gerät oder um OS-9 Terminologie zu verwenden, ein Sequential Character File Device — kurz SCF Device.

Festplatten-Treiber transferieren ganze Datenblöcke und können auch noch gezielt Informationen auf die Festplatte schreiben oder von ihr lesen. Es handelt sich hier also um wahlfreien Blockzugriff. Ein solches Gerät nennt man Random Block File Device (RBF-Device).

²Geräte-Treiber werden von Microware, Inc., dem Hersteller von OS-9, als Teil des Betriebssystems angesehen

<i>File-Manager</i>	<i>Gerätekategorie</i>
SCF	Terminals, Drucker
RBF	Festplatten, Disketten
SBF	Magnet-Tapes
NFM	OS-9 Netzwerk
pipeman	Pipes

Abbildung 1: OS-9 File-Manager

Magnetbänder müssen nochmals anders angesteuert werden. Hier werden zwar auch ganze Datenblöcke übertragen, aber es gibt keinen wahlfreien ("Random") Zugriff mehr, sondern die Datenblöcke werden sequentiell auf das Band geschrieben. Diese Art des Zugriffs ist aus OS-9 Sicht eine weitere Klasse für sich, man nennt solche Geräte Sequential Block File Devices (SBF Device).

Die Unterscheidung in verschiedene Geräte- oder Fileklassen findet aber nicht auf der Stufe des Device Treibers statt, sondern auf der Stufe des File-Managers. Ein File-Manager kennt die Besonderheiten der jeweiligen Gerätekategorie und nimmt die Ansteuerung via Treiber entsprechend vor.

Jede Klasse von Geräten oder Devices hat ihren eigenen File-Manager. So gibt es einen SCF-, einen RBF- und einen SBF-Filemanager, die zum Standardumfang jedes OS-9 Systemes gehören. Weitere, nicht standardmässige Filemanager können zum Beispiel MS-DOS Disketten unter OS-9 verwalten.

Ein weiterer, wichtiger Filemanager ist der Pipe-Manager. Eine Pipe ist im Prinzip ein FIFO-Speicher, den Programme zum Beispiel zur Kommunikation benutzen können. OS-9 kennt "named" und "unnamed" Pipes. "named" Pipes können Sie über das Device `/pipe` ansprechen.

Abbildung 1 zeigt Ihnen die wichtigsten OS-9 Filemanager in einer Tabelle.

Der Aufbau eines OS-9 Treibers

Die Unterscheidung von Gerätekategorien geschieht, wie schon erwähnt, nicht auf Treiberebene; der Treiber muss sich also nur um die effektive Ansteuerung eines Gerätes kümmern.

<i>Routine</i>	<i>Funktion</i>
INIT	Initialisation des Treibers
READ	Lesen von Daten
WRITE	Schreiben von Daten
PUTSTAT	Setzen des Treiber-Status
GETSTAT	Ermitteln des Treiber-Status
IRQSERV	Interrupt Service Routine
EXCSERV	(noch nicht unterstützt)

Abbildung 2: Unterprogramme eines OS-9 Treibers

Ein OS-9 Treiber hat immer denselben Aufbau. Er besteht aus sechs unterstützten und einem reservierten Unterprogramm mit definierten Aufgaben. Abbildung 2 zeigt in Tabellenform die sieben Unterprogramme und Ihre Aufgaben.

Der STATDISP-Treiber

Im folgenden Abschnitt wird der OS-9 Treiber für das Statusdisplay des genannten VMEbus-Computers beschrieben. Das Statusdisplay ist an der Adresse \$FEC04000 Hex anzusprechen, wobei das obere Nibble des Bytes den anzuzeigenden Wert enthalten muss. Der hier vorgestellte Treiber erwartet allerdings entweder einen Wert von \$00 bis \$0F oder den ASCII Code des anzuzeigenden Zeichens ('0' - '9', 'A' - 'F'). Für die ordnungsgemässe Funktion des Treibers ist natürlich ein Deskriptor erforderlich, der unter anderem auch die Hardware Adresse des Displays enthält.

Das Statusdisplay lässt sich hardwarebedingt nicht auslesen, weshalb der Treiber den aktuellen Wert im statischen Speicherbereich ablegt und ihn bei Bedarf als Wert zurückliefert. Es ist aber möglich, dass andere Software, die nicht diesen Treiber verwendet, in der Zwischenzeit das Statusdisplay verändert hat. Der mit READ erhaltene Wert muss also nicht unbedingt stimmen.

Modulheader

```
*****
* OS-9 Device driver for the EUROCOM-5 Status Display
* OS-9 Module header:
```

```

        nam      statdisp
        ttl      Eurocom-5 status display driver

Edition    equ      0
Typ_Lang   set      (Drivr<<8)+0bjct
Attr_Rev   set      ((ReEnt+SupStat)<<8)+0
Stacksize  equ      0
           psect    statdisp,Typ_Lang,Attr_Rev,Edition,0,StatEnt

           use      defsfile

```

Statischer Speicherbedarf

* Static storage offsets

```

           vsect
DispVal    ds.b      1      Current display value
           ends

```

Treiber Einsprung Vektoren

* Entry vector table

```

StatEnt    dc.w      Init
           dc.w      Read
           dc.w      Write
           dc.w      GetStat
           dc.w      PutStat
           dc.w      Term
           dc.w      0 (No exception handler)

```

INIT-Routine

* Initialize status display
*

```

Init:      movea.l V_PORT(a2),a0      get port address
           move.b  #0,(a0)            write '0'

```

move.b	#0,DispVal(a2)	store it for READ
clr.w	d1	no error
rts		return

READ-Routine

* Read: Return the value stored by WRITE or INIT
*

Read:	move.b	DispVal(a2),d0	get value
	clr.w	d1	no error
	rts		return

WRITE-Routine

* Write: Display value and store it in DispVal(a2)
*

Write:	movea.l	V_PORT(a2),a0	get port address
	move.b	d0,d1	save character
	cmpi.b	#\$0f,d0	check if value <= 16
	ble.s	Display	yes, display
	cmpi.b	#'0',d0	value < '0'?
	blt.s	WriteEx	yes, exit
	cmpi.b	#'9',d0	value <= '9'?
	ble.s	WriteAsc	write ASCII char
	bclr	#5,d0	force uppercase
	cmpi.b	#'A',d0	value < 'A'?
	blt.s	WriteEx	yes, exit
	cmpi.b	#'F',d0	value > 'F'?
	bgt.s	WriteEx	yes, exit
	subq.b	#7,d0	subtract 7
	bclr	#5,d1	force uppercase in char
WriteAsc:	subi.b	#'0',d0	make '0' -> #0
Display:	move.b	d1,DispVal(a2)	store character
	lsl.b	#4,d0	shift into upper nibble

```
                move.b  d0,(a0)                display it

WriteEx:        clr.w   d1                    clear carry
                rts
```

GETSTAT-Routine

```
* GetStat:      Get device status
*

GetStat:        cmpi.w  #SS_Ready,d0          ready status?
                bne.s   GetSta10             no
                clr.w   R$d1(a5)            sweep hi word of register
                move.w   #$01,R$d1+2(a5)     return input char count
                rts                          (carry clear)

* return eof status
*

GetSta10:       cmpi.w  #SS_EOF,d0           End of file?
                beq.s   GetSta99            yes, return (carry clear)

* return error for unknow service requests
*

Unknown:        move.w   #E$UnkSvc,d1        Unknown service code
                ori.b    #Carry,ccr         return carry set

* return to caller
*

GetSta99:       rts
```

PUTSTAT-Routine

```
* PutStat: Set device status
*

PutStat:        cmpi.w  #SS_Open,d0          new path opened?
```



```

        beq.s    PutSta_90

        cmpi.w   #SS_Close,d0          path closed?
        bne.s    Unknown

PutSta_90: moveq.l #0,d1                clear carry
        rts

```

TERM-Routine

```

* Term: Terminate status display
*

```

```

Term:    movea.l  V_PORT(a2),a0        get port address
        move.b   #0,(a0)              display '0'
        move.b   #0,DispVal(a2)       store it
        clr.w    d1                   no error
        rts                          return

```

Marc Balmer beschäftigt sich schon seit längerer Zeit beruflich mit OS-9. Seit mehr als zwei Jahren spielt er mit dem Gedanken, eine Zeitschrift herauszubringen, die ganz OS-9 gewidmet ist. Wenn Sie ihm mitteilen möchten, ob dies eine gute Idee war oder nicht, können Sie ihn privat unter der Tel. Nr. +41 61 43 03 85 erreichen.

Erkennen von Exceptions

Carsten Emdé, Wolfgang Ocker

In der Prozessor-Familie MC680x0 von Motorola ist ein sehr mächtiges Instrument zur Fehlerbearbeitung implementiert: das Exception Processing System. Dabei unterscheidet man im Prinzip zwei verschiedene Arten von Exceptions, interne und externe. Interne Exceptions entstehen bei der Bearbeitung bestimmter Instruktionen, z.B. Teilung durch Null oder Angabe einer nicht vorhandenen Instruktion; externe Exceptions können durch Hardware-spezifische Interaktion ausgelöst werden wie z.B. das Adressieren nicht vorhandener Bus-Adressen oder durch Interrupt von I/O-Bauelementen. In jedem Fall von Exceptions reagiert der Prozessor in vergleichbarer Weise auf das Ereignis. Zunächst werden bestimmte Informationen zum Zeitpunkt des Auftretens der Exception auf dem System-Stack abgespeichert. Dabei kann es sich im einfachsten Fall nur um den aktuellen Stand des Status-Registers und des Programm-Zählers (sogenannter short stack frame) handeln, in anderen Fällen, z.B. Bus Error beim MC68030, werden bis zu 90 Byte an diagnostischer Information auf dem Stack gespeichert. In einem zweiten Schritt verzweigt der Prozessor dann an eine Adresse, die aus der sogenannten Vector Base Address und der Art der Exception berechnet wird. Diese Vector Base Address ist bei den Prozessoren MC68000 und MC68008 immer 0, beim MC68010 und dessen Nachfolgetypen besteht ein eigenes Register (das vbr Register), das mit der movec-Instruktion angesprochen werden kann. Tabelle 1 zeigt einen Auszug der Exception-Tabelle.

Am Ende des Exception-Unterprogramms steht normalerweise ein Return-Befehl, nach dem der Prozessor an die Instruktion zurückverzweigt, an der die

<i>Vektor Nummer</i>	<i>vbr-Offset</i>	<i>Bedeutung</i>
0	000	Reset: Interrupt stack pointer
1	004	Reset: Programm-Zähler
2	008	Bus-Fehler
3	00C	Adress-Fehler
4	010	Illegale Instruktion
	usw.	

Abbildung 3: Beginn der Exception-Vektor-Tabelle

Exception aufgetreten ist. In der Regel erfolgt die Weiterbearbeitung des Programms unmittelbar im Anschluss an die fehlerbehaftete Instruktion, bei Bus- oder Adress-Fehlern wird die fehlerhafte Instruktion erneut ausgeführt (retry).

Exception-Unterprogramme im User-Mode

Das Definieren eigener Exception-Unterprogramme im User-Mode ist unter OS-9 recht einfach, da hierfür der spezielle System-Aufruf F\$Trap existiert. Soll zum Beispiel das Auftreten einer Teilung durch 0 von einem Benutzer-Programm abgefangen werden, so braucht man nur eine mit -1 abgeschlossene Tabelle, welche die gewünschte Trap-Nummer und die Adresse des neuen Exception-Unterprogramms enthält, anzulegen und die Adresse dieser Tabelle im Register a1 dem System-Aufruf F\$Trap zu übergeben. Im folgenden Programm-Ausschnitt ist ein kurzes Beispiel für diese Vorgehensweise gegeben:

```
psect _zerdiv_a,0,0,0,0,0

_zerdiv: move.l #0,a0          use current stack in case
                                of exception
        lea.l ExcpTbl(pc),a1   load new exception table
        os9 F$STrap           define it
        rts
ExcpTbl  dc.w T_ZerDiv,ZerDiv--4
        dc.w -1               End of Table
ZerDiv   ...                  here comes the new exception
        ...                  handler
```

Exception-Unterprogramme im System-Mode

Leider besteht die genannte Möglichkeit im System-Mode, z.B. in einem Gerätetreiber, in dieser Weise nicht. Es ist möglich, dass Microware zu einem späteren Zeitpunkt für diese Aufgabe eine Funktion bereitstellt; einen vermutlich für diesen Zweck vorgesehenen siebten Einsprungpunkt gibt es ja bereits jetzt in jedem Treiber. Zur Zeit muss man das Ersetzen des Exceptions-Unterprogramms aber noch "von Hand" vornehmen. Da diese Technik zwar relativ einfach, aber nur spärlich dokumentiert ist, folgt hier eine entsprechende Beschreibung. Besonderes Schwergewicht wird dabei auf die Erkennung eines Bus-Fehlers gelegt. Als

Beispiel soll ein Treiber für eine verschiedenartig konfigurierbare Hardware angenommen werden, bei der vorgegebene Adressen unter bestimmten Umständen physikalisch entweder vorhanden sind oder nicht.

Praktische Realisierung

Der “geschützte Zugriff” auf eine fraglich existierende Hardware-Adresse auf Treiber-Ebene läuft in mehreren Schritten ab:

- Erkennung des Prozessor-Typs, entsprechendes Berechnen und Speichern der Adresse des Standard-Bus-Fehler Exception-Programms
- Sperren aller Interrupts
- Einsetzen des eigenen Exception-Unterprogramms
- Durchführen der kritischen Operation
- Rückinstallieren des Standard Exception-Programms
- Freigabe der Interrupts
- Abfragen eines Fehler-Flags, ob ein Bus-Fehler aufgetreten ist und entsprechende Verarbeitung des Problems

An einer anderen Stelle des Programms muss sich das eigene Exception-Unterprogramm befinden, in dem das genannte Flag im Fehlerfall gesetzt wird. Eine besondere Problematik ergibt sich nun im Zusammenhang mit dem Rücksprung aus dem Exception-Unterprogramm in die normale Weiterbearbeitung. Normalerweise, d.h mit Ausnahme von Bus-Fehler und Adress-Fehler, wird die aktuelle Instruktion zu Ende geführt und ein einheitliches Stack frame angelegt. In diesen Fällen kann man, wie im oben genannten Beispiel im User-Mode, mit dem Return-Befehl aus der Exception-Bearbeitung zurückspringen. Bei Reset, Bus-Fehler und Adress-Fehler wird die aktuelle Instruktion aber nicht zu Ende geführt, und es wird je nach Prozessor- und Exception-Typ ein unterschiedliches Stack frame angelegt. Speziell beim Bus-Fehler werden sehr weitgehende diagnostische Informationen geliefert; der Grund hierfür ist, dass bei Systemen mit virtuellem Memory die gesamte Speicherumschaltung im Bus-Fehler Exception-Unterprogramm verwaltet wird. Wenn man also tatsächlich auch beim Bus-Fehler den üblichen Weg des Rücksprungs aus der Exception-Bearbeitung gehen wollte, müsste man eine detaillierte Prozessor-Erkennung und eigenständige Stack frame-Verwaltung vornehmen. Tabelle 2 zeigt die jeweilige Grösse des Stack frame bei Bus-Error in Abhängigkeit vom Prozessor-Typ.

<i>Prozessor-Typ</i>	<i>Stack frame-Typ</i>	<i>Grösse</i>
68000		14 Byte
68008	--	14 Byte
68010	Format \$8	40 Byte
68012	Format \$8	40 Byte
68020 (Instruktions-Zugriff)	Format \$A	30 Byte
68020 (Instruktions-Verarbeitung)	Format \$B	90 Byte

Abbildung 4: Typ und Grösse des Stack frame beim Bus-Error in Abhängigkeit vom Prozessor-Typ

Wegen dieses relativ komplexen Sachverhalts wird normalerweise davon abgeraten, eigene Exception-Programme im System-Mode zu installieren; in der einschlägigen Literatur findet sich entsprechend der Hinweis, dass Bus- und Adress-Fehler schwere System-Fehler anzeigen und dass es unwahrscheinlich ist, jemals wieder zur normalen Programm-Ausführung zurückkehren zu können.

Da es im vorliegenden Beispiel, Prüfung der Existenz einer physikalischen Adresse, aber zwingend erforderlich war, nach Auftreten eines Bus-Fehlers wieder in den Treiber zurückzuspringen, ohne dass der Benutzer etwas von der Exception-Bearbeitung erfährt, wurde folgende Lösung verwendet: Vor dem kritischen Programmschritt wird die Programm-Adresse, an der die Ausführung nach der Exception fortsetzen soll, sowie der aktuelle Stand des System Stack Pointers gespeichert. Im eigenen Exception-Unterprogramm wird dann nur ein Fehler-Flag gesetzt, der System Stack Pointer zurückgesetzt und nicht mit einem Return-Befehl sondern einfach mit einem Branch-Befehl auf die Rücksprung-Adresse verzweigt. Auf diese Weise umgeht man sämtliche Probleme möglicher Prozessor-Inkompatibilitäten.

Beispiel-Programm

<code>NoIRQ</code>	<code>equ %0000011100000000</code>	Interrupt mask
<code>StdExcpt</code>	<code>ds.l 1</code>	Standard bus error
		exception handler
<code>SysStack</code>	<code>ds.l 1</code>	System stack pointer
<code>SysSR</code>	<code>ds.w 1</code>	Status register
<code>Err102</code>	<code>ds.b 1</code>	Error flag
 <code>EXEntry</code>	 <code>move.b #0,Err102(a2)</code>	 Reset bus error flag
	<code>move.w sr,SysSR(a2)</code>	Save status

	ori.w #NoIRQ,sr	Disable all interrupts
	move.l #0,a3	Default vector base
	move.l D_MPUTyp(a6),d2	
	cmp.l #68010,d2	Any MPU less than 68010?
	bcs.s EX000	Yes, fixed vector base
	movec vbr,a3	No, get vector base
EX000	lea.l T_BusErr(a3),a3	Get Standard bus error exception handler
	move.l (a3),StdExcpt(a2)	Save it
	lea.l Excpt102(pc),a4	
	move.l a4,(a3)	Replace it by our own one
	move.l a7,SysStack(a2)	Save system stack pointer
	---	Do any critical command
EXErr102	move.l StdExcpt(a2),(a3)	Restore standard bus error exception handler
	move.w SysSR(a2),sr	Restore old IRQs
	tst.b Err102(a2)	Was bus error?
	bne.s EXError	Yes
	moveq #0,d1	No error
EXExit	rts	
EXError	---	Provide specific treatment
	move.w #E\$Param,d1	Parameter (or whatever)
	ori.b #Carry,ccr	error
	bra.s EXExit	
Excpt102	move.b #1,Err102(a2)	Set bus error flag
	move.l SysStack(a2),a7	Restore SSP
	bra.s EXErr102	Resume execution

Carsten Emde hat die Entwicklung von OS-9 von Anfang an miterlebt und ist immer noch fasziniert davon. Seine Berufstätigkeit umfasst Entwicklung von OS-9 System- und Anwender Software sowie System-Integration von VMEbus-Rechnern. Er ist privat erreichbar unter der Tel. Nr. +41 1 720 64 47 oder per electronic mail an emde@wopr.open.ch.

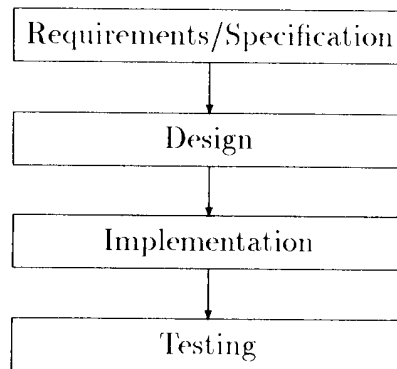
Wolfgang Ocker ist Inhaber der Firma reccoware systems. Er hat vor einigen Jahren OS-9 auf Atari ST Computer portiert und widmet sich seither MGR, der grafischen Benutzeroberfläche, die er für OS-9 verfügbar gemacht und stark erweitert hat. Er ist geschäftlich erreichbar in Unterwittelsbach +49 8251 51 299

A Software Management System

Malindi Lamb

It is generally accepted that a piece of software – or indeed any computer project – undergoes various phases during its life time, generally referred to as the *life-cycle*.

A typical life-cycle might be:



Each of these phases will involve the generation of documents, design diagrams, source code or other data. Most, if not all, of this information is likely to be held on computer. Furthermore, all of the information must be maintained as the software undergoes various revisions for bug fixes and upgrades to the facilities offered.

Maintenance is accepted as being the most expensive element of a software project: the proportion of the cost of maintenance of a project is generally argued to be between 50% and 80%. Given the high cost of maintenance, tools which assist in the automation of maintenance procedures are extremely important and cost-effective.

There are several related tasks which can be automated to assist in software management and maintenance:

- Version Control
- Configuration Management
- Modification Request Management

SMS is a software management system which addresses the problem of automating maintenance procedures. It provides tools to assist in all the above tasks.

Each organisation has its own existing procedures for Version Control, Configuration Management and Modification Request Management: any tools which are to assist in the automation of these tasks must therefore be flexible enough to fit in with existing procedures.

SMS achieves this flexibility by allowing users to set up their own default parameters for all aspects of the system on a project-by-project or even file-by-file basis.

Version Control

Throughout the life cycle of a software project, documents and source code undergo a number of revisions as the requirements change, or the software is developed and updated to incorporate new facilities or correct bugs.

Version Control is the process of maintaining a history of these revisions. Given this revision history such vital questions as:

- What changes were made ?
- Who made the changes ?
- When were the changes made ?
- Why were the changes made ?

can all be answered.

A manual Version Control System can probably answer some of these questions, but the information may not be readily available. All too often the manual Version Control procedures are not enforced, since it is a time consuming and tedious process, so the information available may not be accurate or up-to-date. Furthermore, to maintain manually all revisions of a file or number of files involves keeping complete copies of the files, which is very expensive in storage space.

An automated Version Control system should maintain a revision history of a file by storing only the changes between one revision and the next. This is extremely efficient in terms of storage space, and also enables the Version Control system to show exactly what lines of code changed between one revision and another.

An automated Version Control system should have the capability to maintain not only text files but also data or object files, such as diagram files, data-bases etc. Furthermore, special facilities need to be provided to allow wordprocessor documents to be maintained under the Version Control system.

A fully featured Version Control system also needs to provide facilities for:

- access control
- separate lines of development and the merging of separate lines of development into the main line of development
- automatic revision identification with name, revision number, date etc.
- flexible and powerful report generation

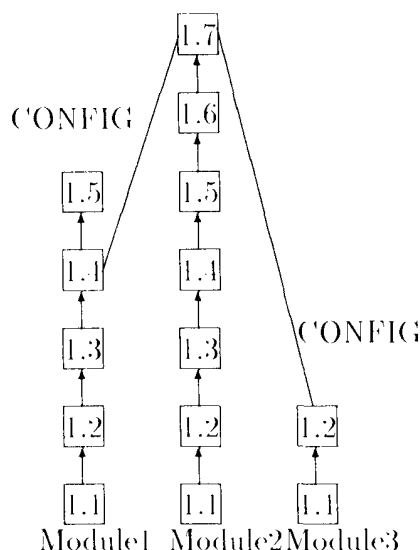
SMS provides all of the above essential features, plus a number of further facilities.

Configuration Management

Configuration Management is the process of managing what makes up a configuration of a piece of software.

This incorporates Version Control for the management of the versions of the modules/programs making up a configuration of a software package.

A Configuration of a software package could be regarded as a slice across the versions of the modules making up the software. A particular configuration (*CONFIG*) may consist of version 1.4 of *module1*, version 1.7 of *module2* and version 1.2 of *module3*, as illustrated in the diagram below.



In order to automate Configuration Management, Version Control tools are required, and these must allow a configuration to be described as a slice across the versions; it should also provide a tool to allow a configuration to be automatically retrieved and built.

SMS provides methods of managing configurations by the use of *symbolic names* to describe a configuration and a *build* utility to generate configurations, maintain them up to date and ensure data integrity during development or maintenance tasks.

Modification Request Management

Modification (or *Change*) *Requests* are bug reports or upgrade requests. Managing these is of great importance in the maintenance process: managers need to know what requests are outstanding, what stage they have reached etc.

Furthermore, there is a direct relationship between the versions of software and the Modification Requests: new versions are in response to a Modification Request.

Most organisations have manual bug reporting and upgrade request procedures, which, as with manual version control systems, may not be accurately main-

tained or kept up-to-date. If they are, then automating the process would save considerable time and probably provide facilities/information difficult to obtain with a manual system.

Every organisation has a different Modification Request system, so an automated system should be flexible to allow it to fit in with existing procedures.

SMS provides a Modification request system which is extremely flexible and fully integrated with the Version Control System.

Modification Request management involves:

- the assignment of Modification Requests to programmers and notifying them that they have been assigned the task of dealing with an MR
- the movement of MRs through various statuses, for example: accepted → fixing → testing → completed → released
- the notification of appropriate personnel on status changes and the reporting of information about MRs

SMS provides all these facilities while still maintaining flexibility and allowing precise statuses used, their names, notifications and so on to be determined by each organisation.

Conclusion

SMS provides a fully integrated Software Management environment, providing facilities for Version Control, Configuration Management and Modification Request Management.

It also provides a menu-driven front-end with full context-sensitive on-line help to allow the benefits of using the system to be achieved with a minimal learning curve.

It is unique in providing a total integrated solution with all the facilities required.

Mrs Malindi Lamb is sales manager at Intasoft Ltd., England. She can be reached during office hours at (0392) 217670.

In eigener Sache

Werbung

OS-9 International stellt für Sie als Firma, die in den Bereichen OS-9, VMEbus, Industrielle Informatik und Automatisierung tätig ist, ein äusserst interessantes Werbemittel dar.

Mit einem Inserat in OS-9 International erreichen Sie direkt Ihr gewünschtes Zielpublikum. Im Gegensatz zu anderen Publikationen wendet sich OS-9 International ausschliesslich an OS-9 Anwender. Eine konzentrierte Werbewirksamkeit ist garantiert.

Verlangen Sie unsere Insertionsbedingungen und die zurzeit gültige Anzeigenpreisliste.

Abonnemente

Die erste Nummer von OS-9 International haben Sie als Probenummer gratis erhalten.

OS-9 International können Sie nur im Abonnement oder einzeln beim Herausgeber beziehen. Die Einzelnummer kostet CHF/DM 8.00, ein Abonnement für sechs Ausgaben kostet CHF/DM 45.00. Universitäten, Hochschulen und Institute können ein Sammelabonnement bestellen. Für den Preis von CHF/DM 100.00 erhalten Sie die nächsten sechs Ausgaben in jeweils zehn Exemplaren. Schicken Sie noch heute eine Karte mit Ihrer Adresse an:

OS-9 International

“Abonnemente”

Hagentalerstrasse 12

CH-4055 Basel

Vergessen Sie nicht zu vermerken, ob Sie ein Abonnement für sechs oder zwölf Ausgaben wünschen. Zusammen mit der nächsten Ausgabe erhalten Sie die Abonnementsrechnung.

Der Einzelverkaufspreis von OS-9 International beträgt im Moment CHF/DM 8.00,

Artikel gesucht

Für die nächsten Ausgaben suchen wir noch Artikel. Publiziert werden Artikel, die OS-9 in irgendeiner Form zum Inhalt haben. Sowohl Programme als auch Berichte über deren Entstehen, spezielle Treiber oder Erfahrungen mit einem bestimmten System sind willkommen. Schicken Sie eine kurze Zusammenfassung Ihrer Arbeit oder den kompletten Artikel zusammen mit einigen Angaben zu Ihrer Person an unsere Adresse.

Artikel können Sie als ASCII-Text in einem der folgenden Diskettenformate zusenden: Alle OS-9 Formate, MS-DOS, Atari ST, Apple Macintosh. Andere Formate nach Absprache.

Impressum

OS-9 International, Magazin für OS-9 Anwender,

herausgegeben von Marc Balmer

Auflage: 1000 Ex.

Redaktion und Gestaltung: Marc Balmer und Carsten Ende

Adresse: OS-9 International, Marc Balmer, Hagentalerstrasse 12, CH-4055 Basel

Electronic Mail: os9-info@urz.unibas.ch

Namentlich gekennzeichnete Beiträge müssen nicht unbedingt der Meinung der Redaktion entsprechen.

Vorschau

In der nächsten Ausgabe von **OS-9 International** werden wir voraussichtlich folgende Themen behandeln:

- **EFFO Forum 21.** Inhalt des EFFO Forum 21
- **Grafische Benutzeroberflächen für OS-9: MGR.** Einer der Punkte, warum OS-9 Mühe hat, sich durchzusetzen, sind die mangelnden Grafikfähigkeiten. Kommandozeilen-Interfaces sind etwas für Experten oder regelmässige OS-9 Benutzer, gelegentliche OS-9 Benutzer wünschen aber eine etwas intuitivere Bedienung des Systems.
- **Terminalauswahl beim Log-In.** Wer oft an verschiedenen Ports einloggt, kennt das Problem: Die Terminalemulation ist in der Regel falsch eingestellt. Das hier vorgestellte Verfahren kann dieses Problem lösen.
- **Wer ist CoVe?.** Der Computer Verein Frankfurt e.V. stellt sich vor.
- **Ein europäisches Netzwerk für OS-9 Benutzer.** Es wird ein Vorschlag gemacht, wie ein gesamteuropäisches Netzwerk für den elektronischen Post-Verkehr und den Austausch von Informationen und Programmen aussehen könnte und wie sich der einzelne Benutzer anschliessen kann. Dieses Netzwerk läuft in der Schweiz bereits erfolgreich.
- **Interprozess-Kommunikation.** Unter OS-9 ist es relativ einfach, zwei oder mehrere Prozesse miteinander kommunizieren zu lassen. Wie's gemacht wird steht in der nächsten Ausgabe.
- **Portierung von UNIX Software auf OS-9.** Welche Probleme stellen sich, wenn man UNIX Software auf OS-9 portieren möchte? Sind der Portierungs-Lust Grenzen gesetzt? Einige Erfahrungen desjenigen Mannes, der den GNU C Compiler für OS-9 portiert hat und ihn der Allgemeinheit zu Verfügung stellt.
- **Pseudo TTYs.** Was PTYs sind und wie sie benutzt werden.

An dieser Stelle möchten wir nochmals darauf hinweisen, dass **OS-9 International** nur im Abonnement erhältlich ist. Schicken Sie uns noch heute eine Karte mit Ihrer Adresse wenn Sie **OS-9 International** weiterhin erhalten möchten (siehe Seite "In eigener Sache" für Details).