



PROCEEDINGS

OF THE INTERNATIONAL SYMPOSIUM'90

ON COMPUTER APPLICATIONS IN INDUSTRIAL AUTOMATION

计算机在工业自动化中的应用 中外研讨会论文集

中国有色金属学会计算机应用学委会

JUNE 1990 BEIJING



PROCEEDINGS
OF THE INTERNATIONAL SYMPOSIUM' 90
ON COMPUTER APPLICATIONS IN INDUSTRIAL AUTOMATION

计算机在工业自动化中的应用
中外研讨会论文集
(下)

中国有色金属学会计算机应用学委会

1990年6月 北京

目 录 (下册)

· S O S 一个用于嵌入系统的小型操作系统 J. E. 塞叶斯.	207
· 用Occam和Pearl进行并行处理——根据实际经验的评价 及对工业和教育后果的展望 H. 戴葛尔曼.	217
· I B M - P C / X T 微机高等数学试题库系统 谭喜薇等.	220
· 计算机在安全系统减程中的应用——事故树的分析 刘波等.	232
· 多种语言、多维报表处理计算机源程序自动生成器A U T O - M + 的设计与实现 田新等.	236
· M O S S 汉字系统的功能及其实现 董俊明.	240
· 3+局域网与D E C N E T 网并存网络的设计与实现 朱治强等.	244
· 《渤铝深加工厂铝门窗C A D、原材料优化、仓库管理系统》分布式 数据库设计 朱治强等.	248
· 基层企业通用电子计算机信息管理系统 刘馥猷.	251
· 用C语言建立可变长记录倒排文件及实现组合查询 ——库存管理中应用一例 范春晓.	253
· 隐含文件名的显示 柏安民.	258
· 在浪潮0520机上实现的图形编辑系统 刘高军.	261
· 使“五笔字型”在长城机的中、高分辨率方式下兼容 柏安民.	264
· 一种用C语言实现的原材料优化下料近似算法 李也白等.	265
· 应用d B A S E - I I I 进行汇编语言的编译 方达上.	270
· 通用型实时点对点通讯系统 于中强等.	272
· 总图运输设计绘图软件包简介 徐已超等.	275
· 俄文—英文兼容磁盘操作系统——一种利用改变系统 硬中断的纯软件方法 于中强等.	278
· 微机辅助设计“计量网络图”综合信息管理系统的研制与应用 谭野勤.	282
· 分析测试数据管理系统开发技术探讨 陈良康等.	285
· 微机在体育田径赛管理中的应用 田永清等.	287
· 面向对象系统S M A L L T A L K / V 的汉化 余社平.	290
· 工业应用程序用的人工智能和软件工程 ——对当前水平的概括看法 许宗德.	292
· 有色金属工业的智能化问题 蔡自兴等.	294
· 改善神经网络系统 δ 学习算法收敛性的一种方法 李晋宏等.	296
· 基于知识的多机器人协调避碰规划 姜志明等.	300

· 用单片机组成的前臂假肢力感反馈电路	魏斌等	306
· 层次型结构的专家系统建造工具	胡仲康等	307
· 冬小麦亲本选配专家系统的设计与实现	陈毅伟等	307
· 钢筋混凝土结构微机辅助设计系统图形处理子系统的开发	蒋芝孟	311
· 生成点集的递归算法及景物仿真	齐东旭	313
· 工业红外热象系统的研究	夏皓如等	315
· 关于多结点样条函数方法与计算机辅助几何造型	徐迎庆	319
· 工业热象的区域识别	王景中	321
· 计算机排课专家系统	吴洁明等	323
· 机器人的最优五次多项式轨迹规划	李宇成	326
· 铝门窗计算机辅助设计系统	陈广胜	328
· 工业机器人手腕伺服传动系统刚性分析	靳桂华	333
· 钢筋混凝土结构微机辅助设计系统(RSCAD)的设计思想	陈善奇	336
· 专家系统不确定知识的表示	高洪深等	338
· 列写节点方程的简捷法	张卫平	340
· 非线性电阻网络分段线性化算法的研究	张卫平	343
· 平面机构运动分析的计算机图解	范珍良等	345
· 齿轮范成的计算机模拟	刘玉茹等	347
· 面向对象系统SMALLTALK及其应用	贾维嘉	348
· 略论回转窑自动控制系统	阎德刚等	351
· 金川公司动力常控参数微机巡回监测系统	杨光寅等	354
· 微机在镍电解净化除钴过程控制的应用	杨光寅等	359
· 开式火焰炉的自动控制	凌昌贵等	365

CONTENTS (Vol. 2)

* SOS -- A Small Operating System for Embedded System .	Jerry E.Sayers	207
* Paralleel Processing with Occam and Pearl -- Evaluations Based on Practical Experience with Outlooks to Industrial and Educational Consequences	H.Deichelmann	217
* IBM PC/XT Microcomputer Advanced Mathematics Examination Queations Base System	Tan Xiwei etc.	220
* Application of Computer in Safty System Engineering -- Fult Tree Analysis	Liu Bo etc.	232
* Auto-M+Source Program Automatic Generator for Multi- Programming Languages and Multi-Dimension Tables Processing : Design and Its Implementation	Tian Xin etc.	236
* A Chinese Character System in MOSS -- Its Function and Realization	Dong Junming	240
* Design and Implementation of Concurrent Network on DECnet -- VAX Network and 3+LAN	Zhu Ziqiang etc.	244
* BLSHCAD System Distribute Database Design	Zhu Ziqiang etc.	248
* A Computer Information Management System Used Generally for Grass-Roots Level Enterprises	Liu Fuyou	251
* Using C-Language Create Invested Files with Various Records and Achieve Organization and Inquirement -- An Example in Stock Management	Fan Chunxiao	253
* Display of Hidden File Names	Bai Anmin	258
* Graph Edit System Realized on LC-0530 Microcomputer	Liu Gaojun	261
* Making the "FSMCC" Operating System Compatible in Middle & High Resolution Mode on Great Wall Computer ...	Bai Anmin	264
* An Algorithm of Optimization and Use of Aluminium Material for Door-Window	Li Yebai etc.	265
* Compile Assembly Language by Use of dBASE-III	Fang Dashang	270
* Common Real-Time Station to Station Communication System	Yu Zhongqiang etc.	272

- * A Brief Introduction to Drafting Software Package
of General Layout Xu Yichao etc. 275
- * Russian & English Compatible DOS Yu Zhongqiang 278
- * The Development and Application of CAD for
"Measuring Network Chart" and CMIS Tian Yeqin 282
- * Studying the Development Technologies of Management
System of Analytical and Testing Data Chen Liangkang etc. 283
- * Application of Microcomputer in the Management
of Track and Competition Tian Yongqing etc. 287
- * Expand Chinese Character for Project-Oriented
Programming SMALLTAK/V Yu Sheping 290
- * Software Engineering and Artificial Intelligence
for Industrial Application Programming
-- An Overview on the state of the Art Henner Schneider 292
- * Study on Some Issues for Intellectualization
of Non-Ferrous Metals Industry Cai Zixin etc. 294
- * A Method of Improving the Convergence of Data Rule
in Neural Network System Li Jinhong etc. 296
- * Knowledge-Based Coordinated Planning
of Multirobot Jiang Zhiming etc. 300
- * A Force-Feeling Feedback Circuit of Prosthetic
Lower Limb Using Single Chip Computer Wei Bin etc. 302
- * The Tool for Establishing Expert Systems
for Hierarchical Structure Hu Zhukang etc. 305
- * Designing and Developing Expert Systems
of Winter Wheat Parents Chen Yiwei etc. 307
- * Development of the Graphic Processing subsystem
Subordinated to the Computer Aided Design
System for the Concrete Structure Jiang Zhimeng 311
- * The Recursive Algorithm of Generating Point Set
and Scene Simulation Qi Dongxu 313

* The Studying of Industrial Infrared Thermal Image System	Xia Haoru	315
* Many-Knot Spline Function Methods and Their Application in CAGD	Xu Yingqing	319
* Region Recognizing of Thermal Image in Industry	Wang Jingzhong	321
* An Expert System for Course Table	Wu Jieming	323
* The Optimal Quintic Polynomial Trajectory Planning for Robot	Li Yucheng	326
* Computer Aided Design System of Aluminium Door-Window	Chen Guangsheng	328
* Sevelomechanism Stiffness Analysis of Industrial Robot Wrist	Jin Guihua	333
* The Design Concept Regarding the Computer Aided Design System for Reinforced Concrete Structure	Chen Shanqi	336
* The Expression of Indefinite Knowledge in Expert Systems	Gao Hongsheng etc.	338
* A Single Algorithm About Construction of Nodal Equation for Linear Networks	Zhang Weiping	340
* Study of Piecewise - Linear Resistive Networks	Zhang Weiping	343
* Computer Graphics for Motion Analysis to the Planar Mechanism	Fang Zhengliang etc.	345
* Computer Simulation of Gear Generating	Liu Yuru etc.	347
* Object-Oriented System -- SMALLTALK and Its Applications	Jia Weijia	348
* On Control System of Kiln	Yan Degang etc.	351
* Power Main Parameter Mobile Monitoring System with Microcomputer in Jinchuan Company	Yang Guangyin etc.	354
* The Application of Microcomputer in Cobalt Removal Process in Purification Section of Nickel Electrolysis Tankhouse	Yang Guangyin etc.	359
* Autocontrol of Open Fired Flame Furnace	Ling Changgui etc.	365

SOS-A SMALL OPERATING SYSTEM FOR EMBEDDED SYSTEMS

Jerry E. Sayers
East Tennessee State University
Johnson City, Tennessee 37614

Abstract

A multitasking operating system for IBM PC or compatible has been developed to provide students with an educational opportunity for hands-on experience with concurrent programming concepts. The operating system includes features which permit users to observe performance a function of the I/O versus compute bound processes. Although other such educational operating system exists, this operating system is limited primarily to process management. The small size of the operating system makes it easy to comprehend in a single term course. It is, however, a powerful real time executive for applications such as small, embedded system.

Introduction

There are many definitions for an operating system. There are many operating systems in use and they accommodate a wide range of applications. However, an operating system may be defined simply as a set of software designed to assist a user of a system in utilizing the computer hardware and its resources. The resources include programs, data, and peripherals.

Despite the variety found in computing hardware and their operating systems, there are many common elements found in all operating systems. The intent of this paper is to give an overview of the basic elements of an operating system and a discussion of the evolution of operating systems development. The driving forces behind that development will be presented. We shall examine the basic types of operating systems and typical applications best suited for them, develop a logical design and give an implementation of a small operating system (SOS) which will run on IBM/PC/XT/AT and 100% com-

patibles.

SOS is a simple, though functional, fixed priority, preemptive operating system (more about that later) suitable for dedicated applications such as might be found in any application requiring real time response to a few multitasked processes. The application used in this paper is that of data collection platform with minimal data management capabilities. The SOS kernel (the basic set of operating system primitive processes) should be adaptable to a wide variety of similar applications. However, the main purpose of this paper is to provide an educational experience with a multitasking operating system augmented by a usable, hands on application. The basic elements found in SOS are also found in all other multitasking systems and should provide a good beginning for students of operating systems.

The Evolution of Operating Systems

Operating System Concepts

Types of Operating Systems

Description of SOS

The logical design of Small Operating System (SOS) will be developed, and the data structure to support an implementation of SOS will be presented. SOS is a fixed priority, preemptive, real time, multitasking system. It consists of a kernel and application tasks. The kernel is the nucleus of an operating system and SOS's kernel will contain system processes, an interrupt subsystem, and support functions such as system start up, system shutdown etc... It would be ideal for very small embedded systems. The system will provide the framework for a sixteen process (or task) system. The Scheduler is allocated to Task 0, the highest priority task in the system. We shall develop a system with, minimally, two tasks, an interrupt subsystem and some support function that make up the kernel. The tasks are the Scheduler and the Work Up task. The interrupt service routines to be written are the Timer handler. A common front of timer interrupt and voluntary suspends will be a part of the interrupt subsystem. The duties of the support procedures will include the follow-

ing:

- .save and load new interrupt vectors
- .restore old interrupt vectors
- .initialize timer, printer, com port, etc...
- .set/reset interrupt masks
- .initialize PCBs
- .provide direct memory video display

Some application processes will be written to demonstrate the use of SOS. These will be 1) a keyboard input manager, 2) a serial communications manager, and all remaining task slots will be filled with CPU intensive tasks to demonstrate the multitasking ability of SOS. A more detailed discussion of the processes, the interrupt subsystem, and the support functions will be given later. First, let us re-explore the concept of a process.

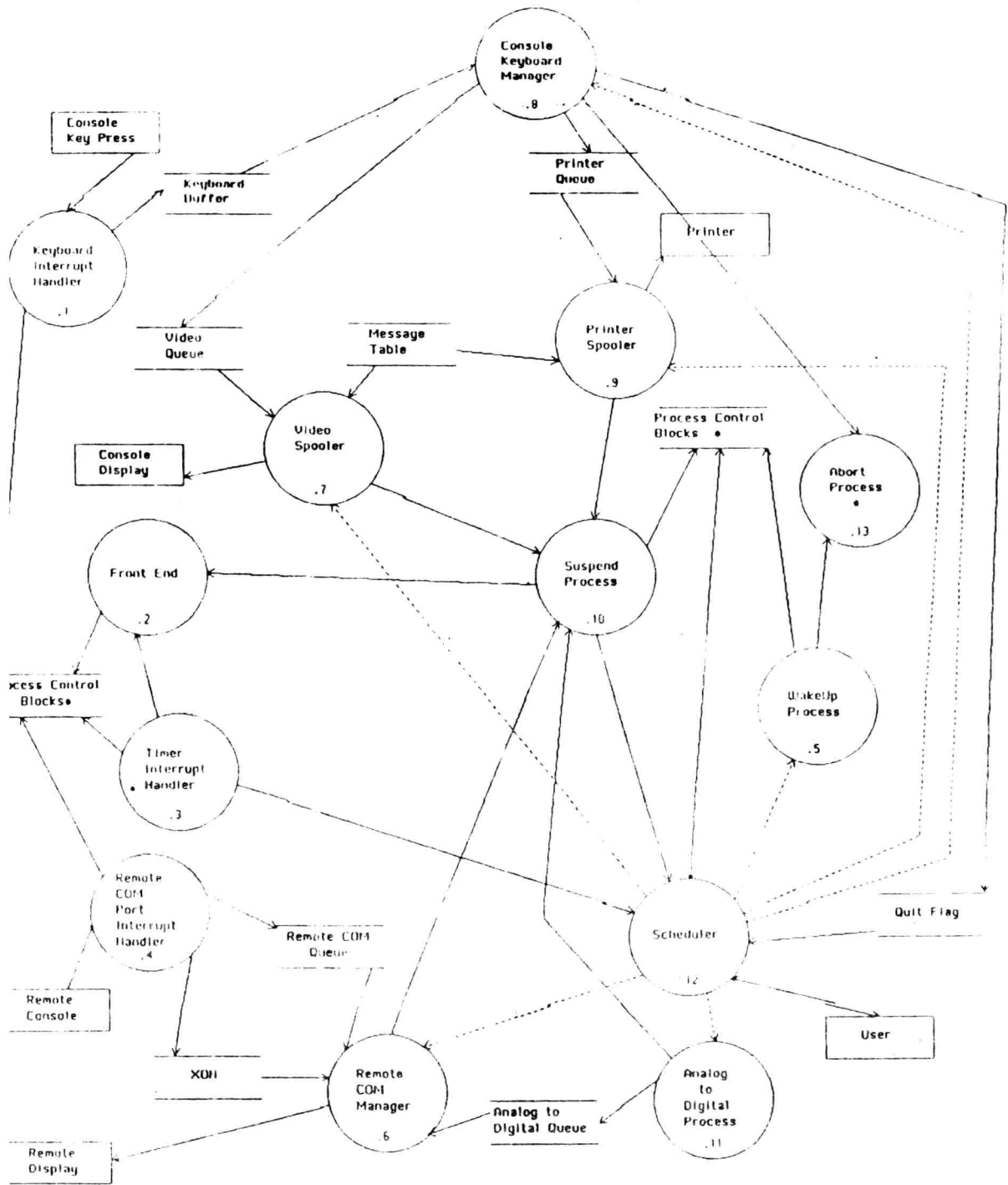
Processes in SOS

The Interrupt Handler/Interrupt Subsystem

Interprocess Communication

Logical Model Diagrams

From the descriptions given above, two diagrams were designed. Figure 2. is a data flow diagram (DFD) which shows graphically how information flows in the SOS environment. In particular, note that the DFD is partitioned into the interrupt subsystem, the multitasking subsystem, and the SOS/MS-DOS domains. Figure 3. is an entity relationship diagram (ERD) which shows the relationship between different objects in the SOS system. These type of diagrams aid in the design of a software system, and the work done developing them is helpful in identifying potential problems before too much effort is committed to the implementation phase of the software life cycle.



*Duplicated on DFD

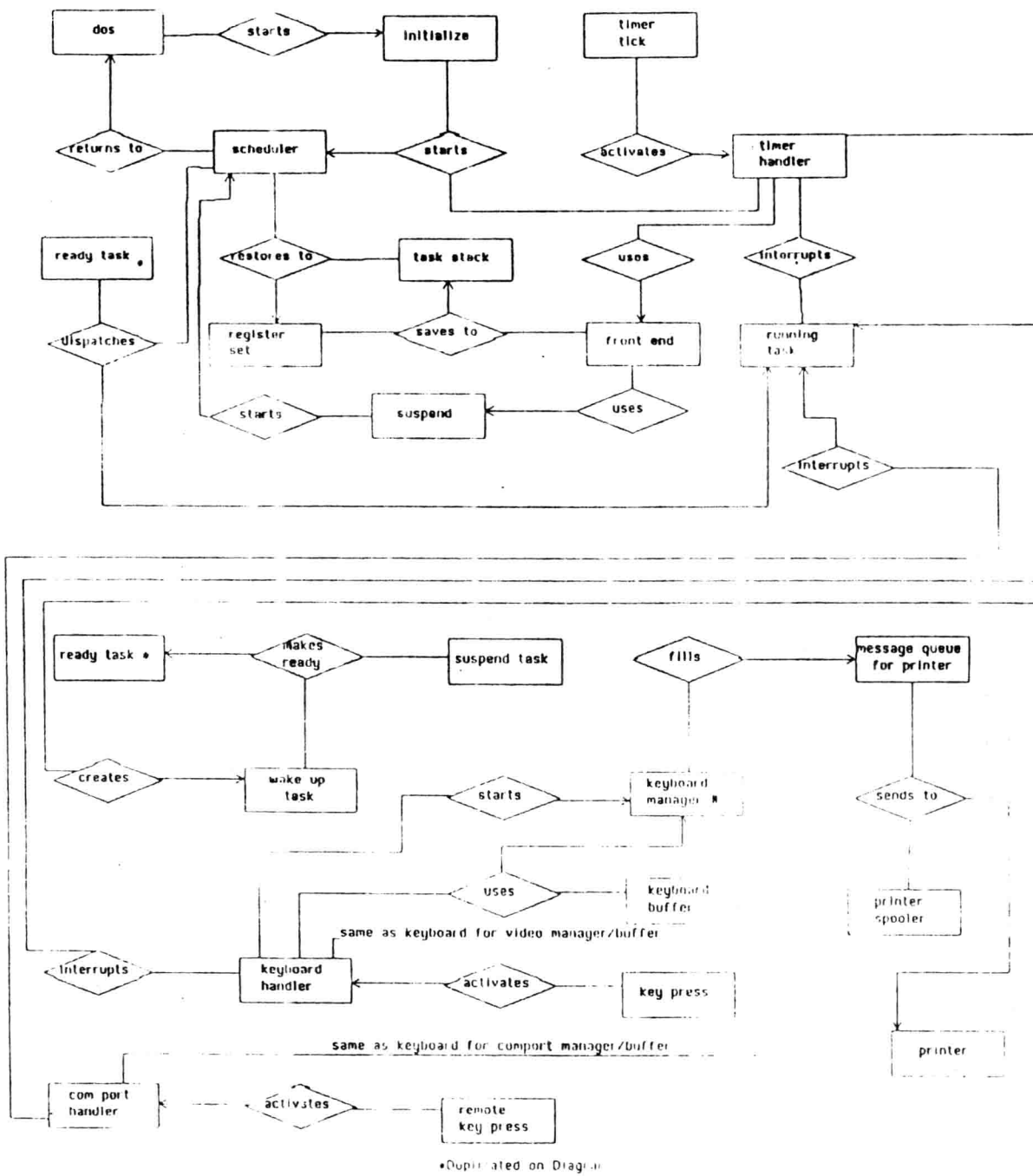


Figure 3 Entity Relationship Diagram for SCS

BIOS/BDOS Problems

Some Other Operating Systems

Basic Design Considerations

The SOS Kernel

Application Processes in SOS

Cyclical Versus Aborted Processes

Queue Management Processes

Special Features of SOS

Two special features were added to SOS for debugging and demonstration of SOS's multitasking capabilities. One feature was fill all unused process slots with a compute bound process and to permit the user to modify the amount of time these processes needed the CPU. The other feature was to have a real time display of the status of each process.

SOS dynamically displays the status of each process as shown in figure 5. This feature requires a lot of overhead, but provides a dynamic view into the states of the operating system processes. The performance loss is acceptable because of the educational benefit derived from this feature. An ancillary benefit of the system status display is its debugging potential. It is easy to verify that transitions of a process occur as expected by viewing the status display. Other features designed to enhance the educational value of the operating system were implemented in some of the user processes. For example, the compute bound processes all are infinite loops which consume CPU cycles for a variable amount of time, and then sleep for a fixed amount of time. The times are directly related to the PID value times a multiplicative factor, thus, the lower the priority, the longer the loop between SUSPEND calls. By pressing the keypad "+" or keypad "-" keys, the multiplicative factor to an

amount that causes the low priority processes to be suspended most of the time. This was done purely for educational purposes. The code used to create several of these processes was

```
PID9    EQU    12          ; Define PIDs for
PIDA    EQU    14          ; processes 9 & A

MAKE_TASK 9          ; Generates a complete PROC.
MAKE_TASK A          ; -place after an ENDP statement.
```

Place in Start Up code to "create" the process on the READY list

```
MOV AX,1              ; Status bit for READY
MOV STATUS[PID9],AX   ; -set for Process 9
MOV STATUS[PIDA],AX   ; -set for Process A
```

Listing 5 is the macro for creating the compute bound processes.

Small Operating System Status

Task	Task
0: NUL-	1: Aborted
2: Aborted	3: Suspended
4: Suspended	5: Suspended
6: Suspended	7: Running
8: Suspended	9: Ready
A: Ready	B: Ready
C: Suspended	D: Ready
E: Suspended	F: Ready

Figure 5. System Status Real Time Display

```

MAKE_TASK      MACRO      N
TASK&N&:       PROC      FAR
TSTART&N&:     ST
               MOV       AX, RUNNING      ; Get PID&N&
               MOV       AH, MULTIPLIER   ; Put multiplier in AH
               MUL        AH
               MOV       CX, AX
TLOOP&N&:      PUSH      CX
               MOV       CX, 4000h        ; Set count (*MULTIPLIER*PID)
TWAIT&N&:      LOOP      TWAIT&N&         ; COMPUTE BOUND !
               POP        CX
               LOOP      TLOOP&N&
               MOV       AX, RUNNING      ; Get PID&N&
               MOV       AH, 55           ; COUNTS FOR 1 second
               MUL        AH              ; Prepare to wait
               PUSHF                     ; PID
               CLI                      ; seconds
               CALL       SUSPEND
               JMP        TSTART&N&       ; Restart task
TASK&N&:       ENDP
               ENDM

```

Listing 5. Macro to Create Compute Bound Processes.

A Sample Application

Conclusion

We have looked briefly at the evolution of operating systems. We have had an overview of the basic components of an operating system as well as a discussion of several flavors of operating systems and the application best suited for them. The purpose of this paper was to lay the ground work for the design, development, and implementation of small, fixed priority, preemptive,

real time operating system named SOS (for Small Operating System). Next the paper presented a design for SOS. SOS is written in 8086 assembly language. Although it was undertaken as an educational project to learn about multi-processing, the resulting product is a very powerful and useful operating system when used within the range of applications for which it was intended.

The reader is referred to the Selected References for further reading. The book by Davis is a user view of operating systems. The book by Mackava et al covers advanced topics like performance measurement and queuing theory. The other books fall into two categories; generic concepts with a hands on operating system developed in the book. Of special interest for this paper is the text by Milenkovic, in which a PC based Real Time operating system is presented. The text by Tanenbaum develops a multiprocessing system Minix which is similar to Unix. Likewise, the text by Comer develops another Unix work alike operating system named Xinu. The others are more generic or are based on a commercial operating system (e.g., Yuen uses the IBM mainframe as his example machine) and represents many of the "classical" operating system books available today. Hopefully, this list will provide you with a good starting place to learn more about operating systems.

SOS — 一个用于嵌入系统的小型操作系统

美国东田纳西州立大学 J. E. 塞叶斯

摘 要

一种用于 IBM PC 或其兼容机的多任务操作系统已经开发出来，它向学生提供学习并行程序概念时亲手实践的环境。该操作系统的特点是允许用户观察 I/O 功能相对于有关计算过程的运行性能。虽然已有其他同类教学用的操作系统，但这些操作系统主要限于进程管理。由于该操作系统的规模小，使它容易在一个学期的课程中来领会。此外它也是一个强有力的实时执行机构，可用于小的嵌入系统中。

标 题

引言，操作系统的演变，操作系统的概念，操作系统的类型，SOS 中的进程，中断处理器/中断子系统，进程通信，逻辑模型图，SOS 的早期发展，BIOS/BDOS 问题，某些其他操作系统，基本设计考虑，SOS 核心，SOS 的应用进程，周期性地撤销进程，排队管理进程，SOS 的特征，采样应用，结论。

结 论

本文简要回顾了操作系统的演变，概述了操作系统的基本组成部分，讨论了操作系统的一些特点及其最适合的应用。本文的目的是为设计、开发和实现一个有固定优先级的小型实时操作系统 SOS（小型操作系统的简称）打下基础。其次，文中介绍了 SOS 的一种设计方法，它是用 8086 汇编语言编写的。尽管它是作为一种学习多任务处理的教学目的而设计的，但其结果的产物则已形成一个在预期应用范围内的强有力的实用的操作系统。