



دانشگاه شهروخ

دانشکده مهندسی

گروه برق

پایان نامه کارشناسی

گرایش: الکترونیک

عنوان: طراحی و پیاده سازی سیستم عامل برای سیستم های سخت افزاری مبتنی بر IBM-PC

استاد راهنما: مهندس احمدی

نگارش: محمد مهدی پور

خرداد ۸۷

یک سیستم عامل...

یک سیستم عامل قدرتمند و مناسب به مصداق باغی است که دارای خاک حاصلخیز و منابع آبی فراوانی در نزدیکی خود است، و برنامه‌هایی که برای یک سیستم عامل ایجاد می‌شوند به مصداق درختان میوه‌ای هستند که در این باغ پرورش داده شده و از محصول آنها استفاده می‌شود. اگر این درختان میوه، میوه‌های مفید، پراستفاده و مقبولی نباشند، چنین باغی بزودی مورد بی توجهی واقع شده و از بین می‌رود...

فهرست

مقدمه ۱

فصل اول - مقدمه‌ای بر سیستم عامل های سیستم های کامپیوتری ۵

مقدمه ۶

وظیفه یک سیستم عامل چیست؟ ۶

تعریف سیستم عامل ۷

عملکردهای سیستم عامل ۸

کنترل فرآیند ۹

مدیریت حافظه ۱۱

مدیریت منابع ذخیره سازی اطلاعات ۱۲

مدیریت سیستم فایل ۱۳

حفاظت و امنیت ۱۴

واسط کاربری ۱۴

مدیریت ورودی/خروجی و ارتباطات ۱۵

تشخیص خطا (Error Detection) ۱۵

فصل دوم - بررسی ویژگی های سیستم های سخت افزاری مبتنی بر IBM-PC ۱۷

مقدمه ۱۸

معماری پردازنده Intel 80x86 ۱۸

۱۸ ثبات های پردازنده.....

۲۰ ساختار حافظه.....

۲۱ دیسک درایوها.....

۲۳ وقفه ها.....

۲۴ وقفه های سخت افزاری.....

۲۴ وقفه های نرم افزاری.....

۲۵ جدول بردار وقفه.....

۲۵ پردازش وقفه ها.....

۲۷ فصل سوم - معرفی سیستم عامل MiniTAK.....

۲۸ مقدمه.....

۲۹ قابلیت ها و امکانات پیاده سازی شده.....

۳۰ کاربردهای سیستم عامل.....

۳۲ فصل چهارم - بررسی ساختار داخلی سیستم عامل MiniTAK.....

۳۳ مقدمه.....

۳۶ بررسی ساختار کلی سیستم عامل.....

۳۷ راه انداز سیستم.....

۳۸ مدیریت حافظه.....

۳۹ کنترل فرآیند.....

بارکننده برنامه ۴۱

مدیریت منابع ذخیره سازی اطلاعات ۴۲

مدیریت ورودی/خروجی و ارتباطات ۴۲

واسط کاربری ۴۳

هسته سیستم عامل ۴۳

فصل پنجم - بررسی جزئیات داخلی فایل های سورس سیستم ۴۵

مقدمه ۴۶

فایل MINIO.SYS ۴۶

فایل MINITAK.SYS ۵۰

فایل MINIMAN.COM ۵۵

فصل ششم - ابزارهای برنامه نویسی و اشکالزدایی ۵۸

مقدمه ۵۹

ابزارهای برنامه نویسی ۵۹

MASM (Microsoft Macro Assembler) ۵۹

NASM (Netwide Assembler) ۶۰

ابزارهای اشکالزدایی ۶۰

فایل Debug.inc (بخشی از پروژه) ۶۰

Microsoft Code View ۶۱

۶۲ Borland Turbo Debugger

۶۳ برنامه DEBUG

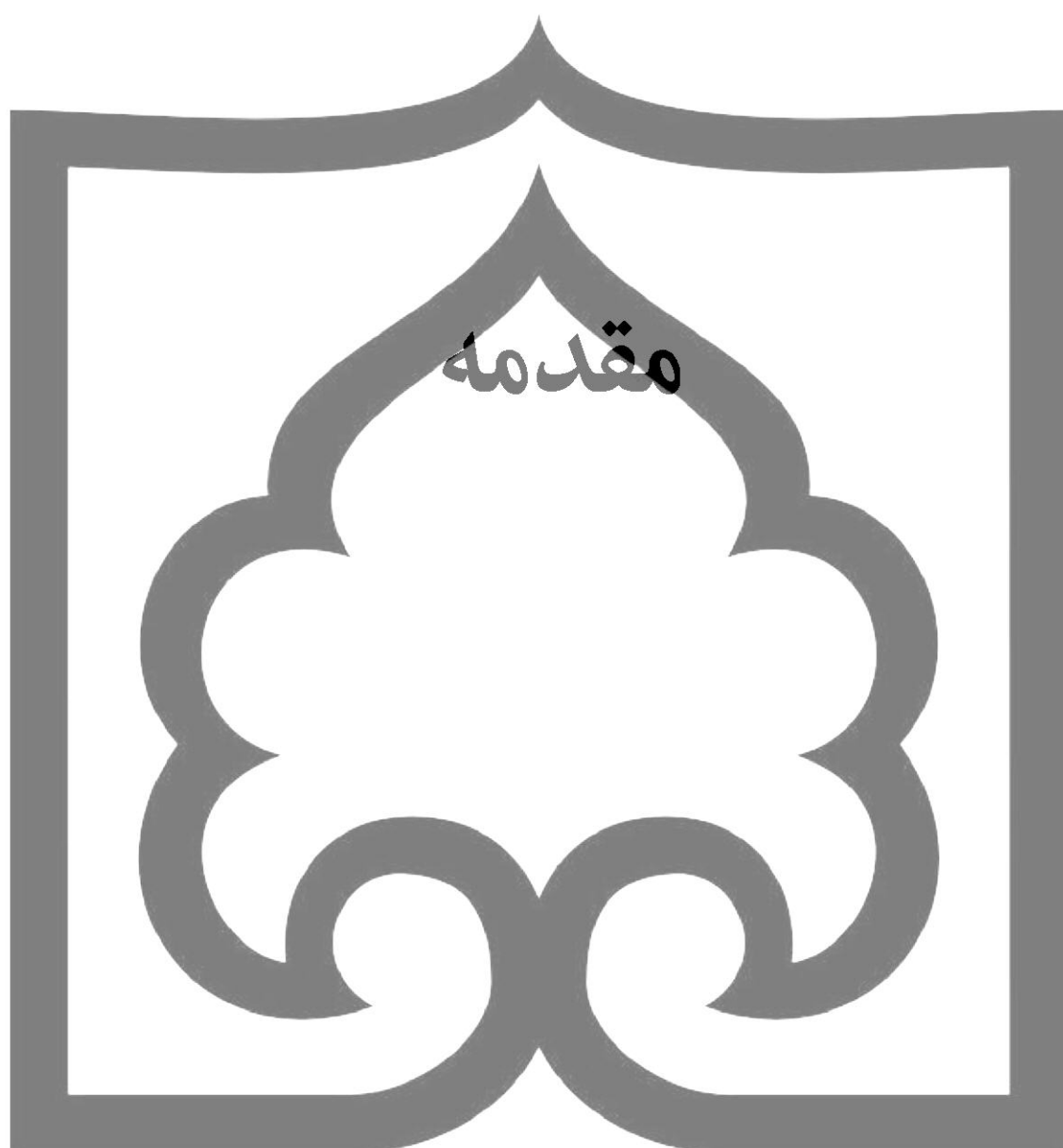
۶۳ Microsoft Virtual PC

۶۴ VMware Workstation

۶۶ ضمایم

۶۷ ضمیمه الف- مقاله ارائه شده به یازدهمین کنفرانس دانشجویی مهندسی برق ایران

۷۷ مراجع



بطور معمول در سیستم‌های قابل برنامه ریزی کوچک نظیر میکروپروسورها و میکروکنترلرها که حجم حافظه برنامه ریزی و منابع سخت‌افزاری محدود است، برنامه‌های کاربردی بگونه ای نوشته می شوند که بطور کامل کنترل کلیه منابع و مدیریت آن‌ها را در دست گرفته و تمام توابع موردنیاز را به همراه خود فراهم نموده باشند. این روش برنامه نویسی برای این سیستم‌ها، در اکثر کاربردهایی که نیازی به تغییرات زود به زود قابلیت‌های برنامه نباشد، دارای کارایی است.

با این وجود، در کاربردهای پیچیده که به تغییر قابلیت‌های برنامه و یا حتی تغییر بستر سخت‌افزاری نیاز است، پروسه برنامه نویسی به این دلیل که می بایست توابع اصلی سیستم با پیکربندی جدید برنامه و یا بستر سخت‌افزاری جدید سازگار شوند، وقتگیر و پیچیده‌تر خواهد شد و هزینه‌های زمانی آن افزایش خواهد یافت. علاوه بر آن به دلیل عدم وجود یک استاندارد یا کتابخانه مرجع در ساخت این برنامه‌ها، برنامه نویسان مختلف ممکن است برخی توابع اصلی را به گونه‌های مختلف و احیاناً غیرکارا پیاده سازی نمایند.

مهمترین نکته قابل توجه در این مسأله زمانی آشکار می شود که به تغییر بستر سخت‌افزاری برنامه‌های ایجاد شده برای یک سیستم نیاز باشد. در اینگونه موارد می توان از طریق سازگار کردن بستر نرم‌افزاری (سیستم عامل) این برنامه‌ها با سخت‌افزار جدید و پیاده سازی دوباره توابع اصلی و هسته سیستم عامل، آن‌ها را در بستر سخت‌افزاری جدید بدون هیچگونه مشکلی بکار برد.

با توجه به اینکه سیستم‌های سخت‌افزاری مبتنی بر ریزپردازنده‌ها و سخت‌افزارهای IBM-PC بطور گسترده‌ای در پروژه‌ها استفاده می‌شوند، سیستم عامل و یا شبه سیستم عامل کنترل‌کننده این سیستم‌ها یکی از مباحث اصلی در ساختار این پروژه‌ها می‌باشد. همچنین در پروژه‌هایی که حجم پردازش عملیاتی

بسیار زیاد می‌باشد، یکی از مهمترین معیارها در طراحی سیستم عامل برای این سیستم‌ها داشتن حداکثر کارایی و استفاده حداکثر از منابع پردازشی پردازنده می‌باشد. در اینگونه پروژه‌ها می‌توان با ادغام برنامه اصلی در سیستم عامل به این هدف دست یافت که برای این منظور می‌بایست سورس سیستم عامل در

دسترس باشد. بنابراین، سخت‌افزار IBM-PC به عنوان یک سخت‌افزار نمونه که دارای امکانات تقریباً کامل برای پیاده‌سازی یک سیستم عامل نمونه می‌باشد، انتخاب شده است.

در این پروژه، هدف، طراحی یک سیستم عامل Open Source، کاربردی و با اندازه کوچک می‌باشد. ویژگی اصلی این سیستم عامل پشتیبانی کامل آن از برنامه‌های اجرایی سیستم عامل MS-DOS است،

زیرا با وجود برنامه‌های کاربردی بسیاری که برای سیستم عامل DOS ایجاد شده و کاربران بسیاری نیز به آن عادت کرده‌اند، یکی از راه‌های کاربردی کردن این سیستم عامل پشتیبانی آن از برنامه‌های DOS است. از طرف دیگر، با توجه به اینکه با ساخت هر سیستم عامل، باید ابزارها و برنامه‌های لازم برای تولید و توسعه نرم‌افزارهای کاربردی برای آن تولید شود، با سازگار کردن این سیستم عامل با سیستم عامل DOS، هزینه‌های تولید کامپایلرها و ابزارهای توسعه نرم‌افزار از مراحل ساخت این سیستم عامل حذف می‌شود، زیرا محیط‌ها و ابزارهای برنامه‌نویسی و اشکال‌زدایی متنوع و قدرتمندی تحت سیستم عامل DOS ایجاد شده‌اند.

همچنین با توجه به اینکه سیستم عامل‌ها جزء برنامه‌های سیستمی شمرده می‌شوند و در نزدیکترین سطح به سخت‌افزار سیستم قرار دارند، پیاده‌سازی آن‌ها مستلزم داشتن دانش دقیق و کامل از سخت‌افزار و استانداردهای نرم‌افزاری موجود برای این خانواده از سیستم‌ها می‌باشد. به عنوان مثال، با توجه به اینکه دیسک درایو یکی از اجزای اصلی موجود در یک سیستم سازگار با IBM-PC برای ذخیره اطلاعات است، می‌بایست نحوه عملکرد دقیق آن کاملاً شناخته شود. به عنوان مثالی از استانداردهای نرم‌افزاری نیز می‌توان به روش‌های ذخیره فایل‌ها بر روی دیسک درایو اشاره کرد که می‌بایست سازگار با یکی از سیستم فایل‌های مرسوم باشد. علاوه بر آن، علی‌رغم اینکه کتاب‌های متعددی در زمینه مفاهیم سیستم عامل و طراحی ساختار داخلی آن‌ها وجود دارند که فناوری‌های مختلف بکار رفته در آن‌ها را توضیح می‌دهند، هیچکدام از آن‌ها در مورد نحوه پیاده‌سازی عملی سیستم عامل‌ها اطلاعات کاملی در اختیار

قرار نمی‌دهند. به همین دلایل، بیش از ۵۰٪ از مراحل ساخت این سیستم عامل پژوهشی بوده و بر روی فناوری و دانش پیاده‌سازی عملی سیستم عامل واقعی و نیز ویژگی‌های سخت‌افزارهای سازگار با IBM-PC می‌باشد.

در مرحله پیاده‌سازی هدف این بوده است که سیستم عامل قابلیت اجرا بر روی پردازنده‌های ۸۰۸۸ را دارا باشد که در اینصورت سازگاری آن با پردازنده‌های قوی‌تر تضمین می‌شود. این سیستم عامل، یک سیستم عامل ۱۶ بیتی بوده، بطور ۱۰۰٪ به زبان اسمبلی نوشته شده و مجموعه سخت‌افزاری نمونه که این سیستم عامل بر روی آن تست می‌شود، کامپیوتر شخصی (PC) خواهد بود که در اینصورت سازگاری آن با زیرمجموعه‌های این سخت‌افزار نمونه نیز تضمین خواهد شد. سورس سیستم عامل پیاده‌سازی شده برای تولید نسخه‌های سفارشی و ترکیب آن با برنامه‌های دیگر نیز در دسترس خواهد بود. نسخه فعلی و نهایی این سیستم عامل بالغ بر ۲۰۰۰۰ خط برنامه اسمبلی است.

سیستم عامل پیاده‌سازی شده بنام سیستم عامل MiniTAK نامگذاری شده است. همچنین لازم به ذکر است که پیاده‌سازی این سیستم عامل بخشی از یک پروژه طراحی سیستم عامل می‌باشد که براساس آن، مقاله‌ای برای یازدهمین کنفرانس دانشجویی مهندسی برق ایران ارائه شده است. مقاله مذکور ضمیمه پایان‌نامه شده است.

فصل اول

مقدمه‌ای بر سیستم
عامل‌های سیستم‌های
کامپیوتری

مقدمه

یک سیستم عامل به عنوان واسطی بین کاربران سیستم و سخت افزار سیستم کامپیوتری عمل می کند. وظیفه یک سیستم عامل فراهم کردن محیطی است که کاربران بتوانند بر راحتی برنامه های مورد نظر خود را اجرا کرده و کارهای خود را انجام دهند.

سیستم عامل نرم افزاری است که سخت افزار سیستم را مدیریت می کند. سخت افزار سیستم باید مکانیسم ها و امکانات لازم برای عملکرد صحیح و پایدار سیستم کامپیوتری را تضمین نموده و امکانات لازم برای انجام عملیات های مختلف را با فراهم کردن راه های ارتباطی استاندارد برای نرم افزار، در اختیار سیستم عامل قرار دهد.

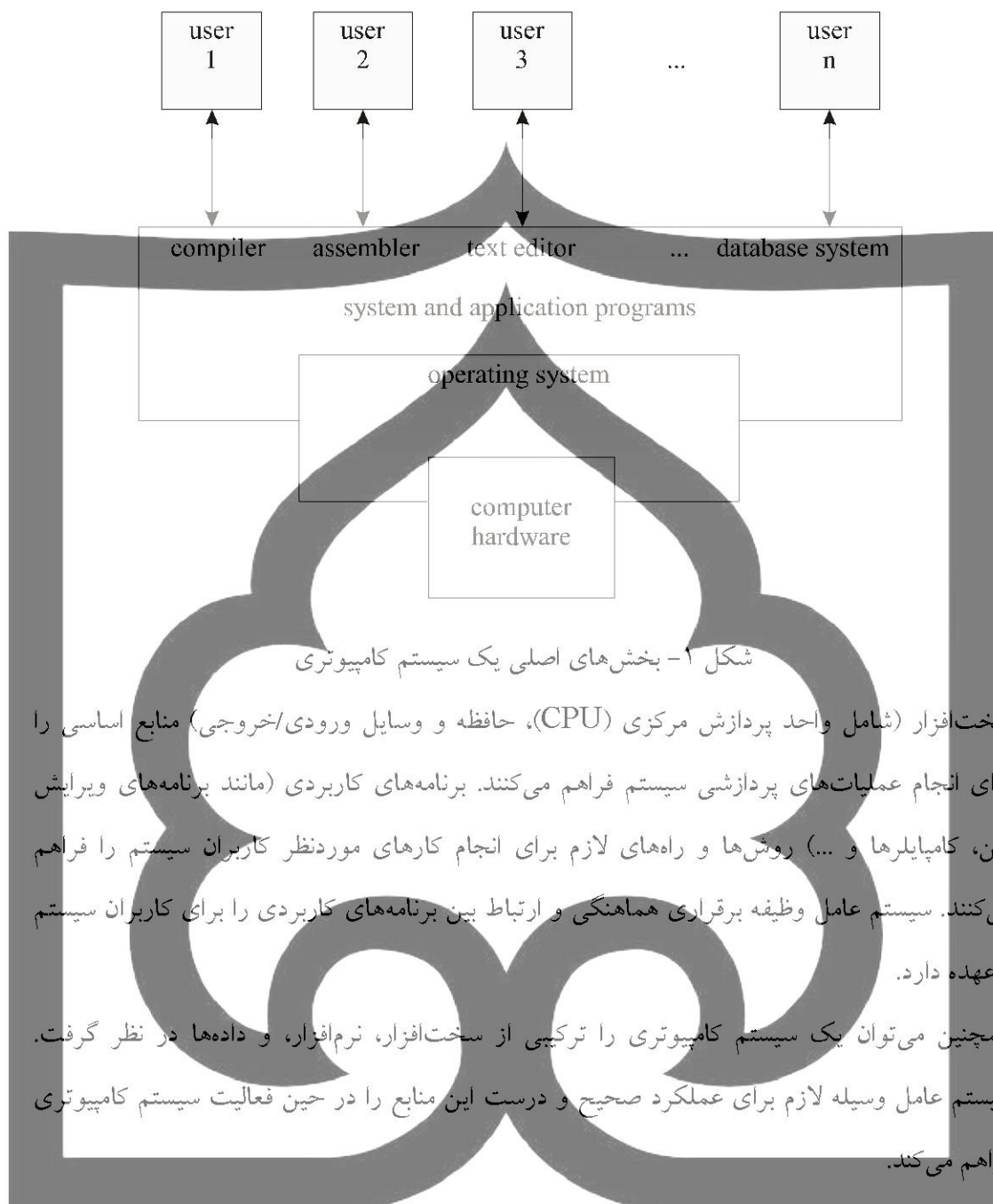
سیستم عامل ها از نظر ساختار داخلی به میزان زیادی با یکدیگر تفاوت دارند. طراحی یک سیستم عامل جدید عملیات سنگینی است. یک نکته مهم در طراحی سیستم عامل این است که باید اهداف و کاربردهایی که از سیستم عامل مورد نظر انتظار می رود، قبل از شروع مراحل طراحی و ساخت بطور دقیق تعریف شود. تعریف دقیق این اهداف و وظایف، شروط اساسی در انتخاب الگوریتم ها و استراتژی های لازم در طراحی را شکل می دهند.

از آنجا که سیستم عامل، برنامه های بزرگ و پیچیده است، می بایست مرحله به مرحله و بخش به بخش ساخته شود. هر کدام از این بخش ها باید بخش کاملاً تعریف شده و مشخصی از پیکره کلی سیستم عامل بوده، توابع و عملکردهای آن با دقت تعریف شده باشند.

در این فصل، یک دید کلی از اجزای اصلی سازنده یک سیستم عامل ارائه می شود.

وظیفه یک سیستم عامل چیست؟

یک سیستم کامپیوتری را می توان به چند بخش اصلی تقسی کرد: سخت افزار، سیستم عامل، برنامه های کاربردی، و کاربران (شکل ۱).



تعریف سیستم عامل

بطور عمومی هیچ تعریف کاملاً دقیق از سیستم عامل وجود ندارد. سیستم عامل‌ها به این دلیل وجود دارند که می‌توانند راه‌های منطقی و مقرون به صرفه‌ای برای حل مسائل و مشکلات موجود برای راه‌اندازی یک سیستم کامپیوتری قابل استفاده را فراهم نمایند. هدف اصولی یک سیستم کامپیوتری

اجرای برنامه‌های کاربران سیستم و راحت‌تر کردن انجام وظایف مورد نیاز کاربران می‌باشد. برای رسیدن به این منظور، سخت‌افزار کامپیوتر ایجاد شده است. اما چون بکارگیری مستقیم سخت‌افزار آسان نیست، برنامه‌های کاربردی مختلف ایجاد شده‌اند. این برنامه‌ها برای انجام وظایف مورد نظر کاربران نیاز به پشتیبانی از برخی عملکردهای پایه‌ای، مانند کنترل تجهیزات ورودی/خروجی متصل به سیستم، که در ارتباط با سخت‌افزار سیستم است، دارند. توابع و سرویس‌های لازم برای انجام این عملیات‌ها و استفاده از منابع سیستم در قالب نرم‌افزار بزرگی بنام سیستم عامل گرد هم آمده‌اند.

عملکردهای سیستم عامل

سیستم عامل‌های مدرن برای عملکرد خود از حالت وقفه‌ای استفاده می‌کنند. در این حالت، اگر هیچ فرآیندی در حال اجرا نباشد، هیچ وسیله خارجی نیاز به کنترل نرم‌افزاری نداشته باشد و هیچ کاربری نیز در حال کار با سیستم نباشد، سیستم عامل به حالت انتظار رفته و منتظر وقوع یک رخداد باقی می‌ماند. سیستم عامل معمولاً در قالب یک وقفه سخت‌افزاری یا نرم‌افزاری از این رخدادها باخبر می‌شود. طبیعت وقفه‌ای یک سیستم عامل ساختار عمومی سیستم را تعریف می‌کند. برای هر نوع وقفه، قطعه برنامه‌ای در داخل سیستم عامل وجود دارد که مشخص می‌کند چه اعمالی در پاسخ به وقفه مورد نظر باید انجام گیرد. به این قطعه کدها روال‌های سرویس وقفه^۱ گفته می‌شود که مسئول پاسخ‌دهی به وقفه مربوطه است. از آنجا که سیستم عامل و کاربران، منابع نرم‌افزاری و سخت‌افزاری را به اشتراک می‌گذارند، باید به طریقی این اطمینان بوجود آید که وقوع خطا و اشکال در یکی از برنامه‌های در حال اجرای کاربر، فقط اجرای برنامه‌ای را که مشکل در آن بوجود آمده، مختل می‌نماید. زیرا در حالت اشتراک منابع، این احتمال می‌تواند وجود داشته باشد که برنامه‌های در حال اجرای دیگر نیز با وقوع اشکال در عملکرد یک برنامه، تحت تأثیر قرار گیرند. بطور مثال، اگر برنامه‌ای وارد یک حلقه بی‌نهایت شود، این حلقه اجرای بی‌نهایت می‌تواند جلوی اجرای صحیح بسیاری از فرآیندهای در حال اجرای دیگر روی سیستم را بگیرد. خطاهای بغرنج‌تری که بسختی قابل شناسایی هستند، در سیستم‌های برنامه‌سازی چندگانه^۲ ممکن

^۱ Interrupt Service Routine

^۲ Multiprogramming System

دانشجویان محترم:

جهت دسترسی به متن کامل پایان نامه ها به کتابخانه دانشکده مهندسی و یا آزمایشگاه پروژه گروه برق مراجعه فرمایید.

- [1] A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts*, John Wiley & Sons, 2005
- [2] W. Stallings, *Operating Systems: Internals and Design Principles*, 5th edition, Prentice Hall, 2005
- [3] A.S. Tanenbaum, *Modern Operating Systems*, Prentice Hall, 2001
- [4] J. Bacon, T. Harris, *Operating Systems: Concurrent and Distributed Software Design*, Addison Wesley, 2003
- [5] M.A. Mazidi, J.G. Mazidi, *80x86 IBM PC and Compatible Computers: Assembly Language, Design, and Interfacing*, Volume I and II, 2nd edition, Prentice Hall, 1997
- [6] J. Uffenbeck, *The 80x86 Family: Design, Programming, and Interfacing*, 3rd edition, Prentice Hall, 1997
- [7] J. P. Royer, *Handbook of Software and Hardware Interfacing for IBM PCs*, Prentice Hall, 1986
- [8] M. Tischer, *PC Intern: PC System Programming*, Volume I, II, III, Abacus, 1992
- [9] P. Norton, *Inside the IBM PC and PS/2*, Brady, 1991
- [10] P. Norton, S. Holzner, *Advanced Assembly Language*, Brady, 1991
- [11] P. Norton, *Peter Norton's Assembly Language Book for the IBM PC*, Brady, 1989
- [12] R. S. Lai, The Wait Group, *Writing MS-DOS Device Drivers*, Addison Wesley, 1992
- [13] P. Norton, *Peter Norton's Complete Guide to DOS 6.22*, Sams, 1994