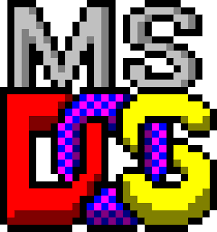
**سیستم عامل**

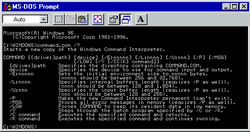
**انواع سیستم عامل از نظر ساختاری**

**یکپارچه (ساده)**

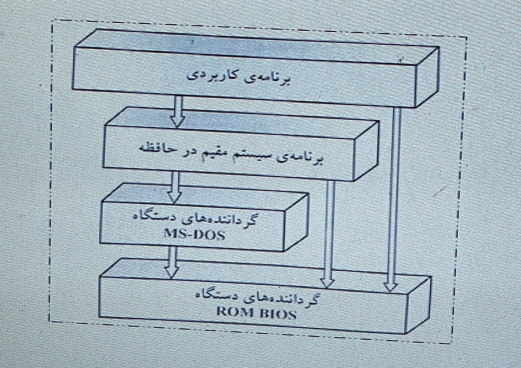
ساده ترین ساختار برای سیستم عامل است در این ساختار واسط ها و سطوح عملکرد به خوبی از هم تفکیک نشده اند و برنامه های کاربردی می توانند به روال i/o(ورودی و خروجی) دستیابی داشته باشند و مستقیما بر روی مانیتور یا دیسک بنویسند

اغلب سیستم عامل های تجازی. ساختار های خوش تعریفی ندارند غالبا چنین سیستم هایی به صورت سیستم هال کوچیک ساده و محدود شروع به کار می کنند و سپس نسبت به حوزه اصلی خود رشد می کنند





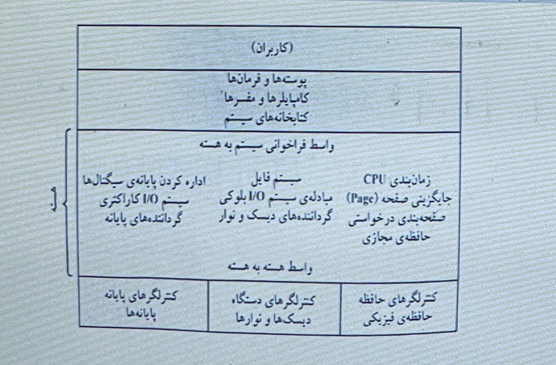
MS DOS مثالی از چنین سیستمی است . این سیستم عامل نوشته شده تا بیشترین قابلیت را در کمترین فضا فراهم سازد و در نتیجه به دقت به پیمانه ها تقسیم نشده است



در MS-DOS واسط ها و سطوح عملکرد به خوبی تفکیک نشده اند برای مثال برنامه های کاربردی قادرند به روال های i/o پایه نیز دستیابی داشته باشند یا مستقیما در نمایشگر و گرداننده های دیسک بنویسند این آزادی ms-dos را در مقابل برنامه های مضر آسیب پذیز می سازد و در نتیجه موجب از کار افتادم سیستم می شود

چون در intel 8088 که سیستم عامل MS-Dos برای آن نوشته شد فاقد حالت دوگانه و فاقد حفاظت سخت افزاری است طراحان ms-dos هیچ انتخابی نداشتند و سخت افزار را دسترس پذیر رها کردند

مثال دیگر ساختار محدود سیستم عامل یونیکس اولیه است همانند MS-DOS یونیکس ابتدا توسط امکانات سخت افزاری محدود شد یونیکس شامل دو بهش بود هسته (کرنل - kernel) و برنامه های سیستم . هسته به مجموعه ای از واسط ها و گرداننده های دستگاه تبدیل شد که طی چندین سال اضافه شدند و بسط یافتند . سیستم عامل های قدیمی یونیکس را می توان لایه ای در نظر گرفتدر شکل زیر امده است هر چیز موجود در در پایین واسط فراخوان سیستم و بالای سخت افزار فیزیکی هسته است

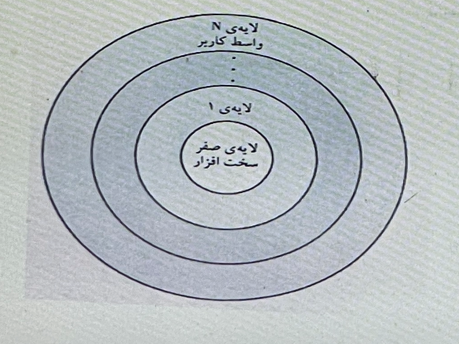


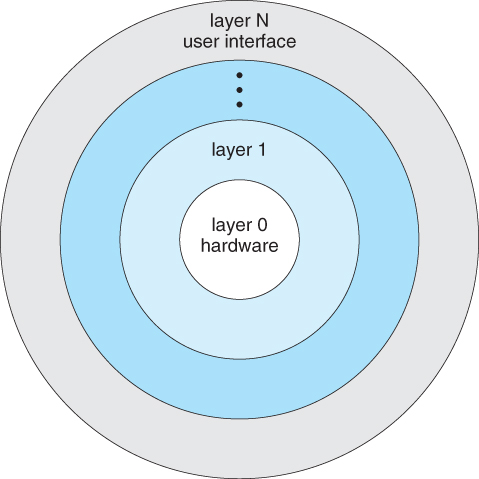
هسته از طریق فراخوان های سیستمی سیستم فایل زمانبندی پردازنده مدیریت حافظه و سایر سیستم عامل را فراهم می سازد .جمع شدن امکانات (عملکرد های) زیاد در یک سطح اشتباه است پیاده سازی و نگهداری این ساختار یکپارچه دشوار بود هر چند که امتیاز خاصی نیز داشت: در واسط فراخوان سیستم یا در ارتباطات داخل هسته سرباز بسیار کمی وجود دارد

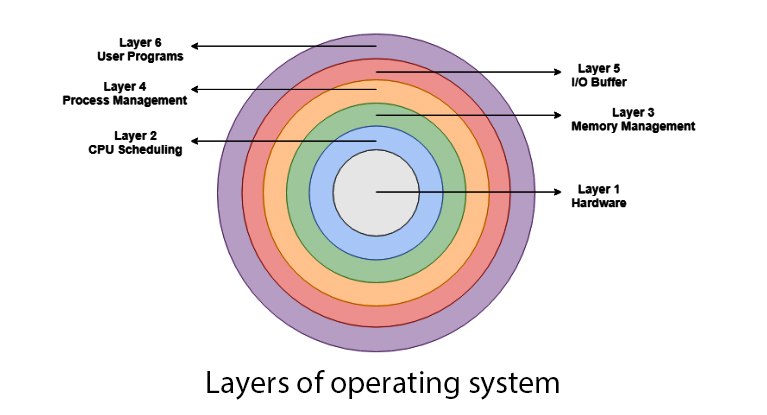
**لایه ای**

با پشتیبانی سخت افزاری مناسب سیستم عامل ها می توانند به مولفه هایی تبدیل شوند که نسبت به سیستم عامل های یونیکس و ms-dos اولیه کوچک تر و مناسب تر باشند در این صورت سیستم عامل روی کامپیوتر و برنامه های کاربردی که از آن استفاده می کنند کنترل خیلی بیشتری دارد پیاده ساز ها, آزادی بیشتری در تغییر عملکرد داخلی سیستم و ایجاد سیستم عامل های پیمانه ای دارند **پنهان سازی اطلاعات به پایین** . ویژگی ها و عملکرد کلی تعیین به مولفه هایی تقسیم می شوند **پنهان سازی اطلاعات** نیز مهم است

سیستم عامل به چند لایه (سطح ) تبدیل می شود لایه پایینی (لایه شماره0 ) سخت افزار است لایه ی بالایی (لایه شماره N) واسط کاربر استاین ساختار لایه ای در شکل زیر امده است







نمونه ای از لایه ی سیستم عامل – مثلا لایه ی M شامل ساختمان داده