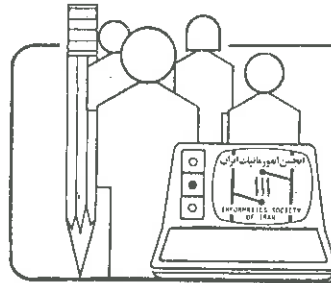


گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران



سال پنجم - شماره ۴
شماره پیاپی ۴۶
امرداد ۱۳۶۲
تک شماره ۵۰ ریال - اشتراک سالانه ۵۰۰ ریال



اخبار انجمن

تغییر در ترکیب هیأت اجرایی

آقای فرهاد صدی اعتمادی، یکی از اعضای هیأت اجرایی انجمن، اخیراً "از سمت خود استعفا داده اند. از آتجائی که در انتخابات قبلی انجمن (آبان ماه ۱۳۶۰) سه عضو علی البدل برای هیأت اجرایی تعیین شده بودند، آقای حمید خورسند رحیم زاده که در بین اعضای علی البدل بیشترین تعداد آراء را داشتند به عضویت هیأت اجرایی انجمن درآمدند.

نشریات واصله به کتابخانه انجمن

نشریات زیر طی سه ماهه اول سال ۱۳۶۲ به کتابخانه انجمن واصل شده اند:

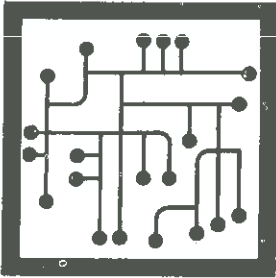
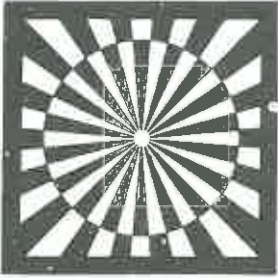
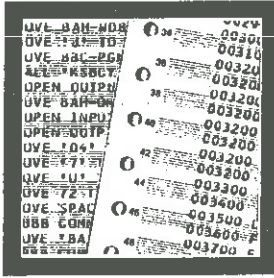
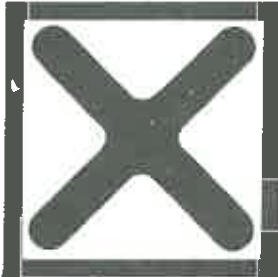
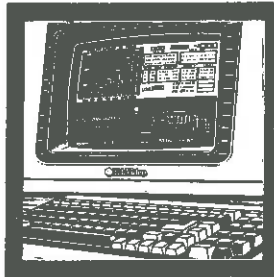
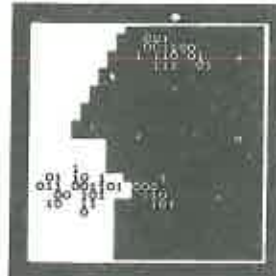
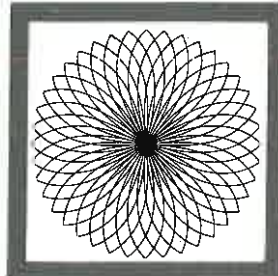
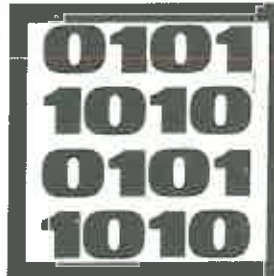
- ۱ - کد کامپیوتری جدول آب و بخار، شرکت توانیر، اردیبهشت ۱۳۶۲.
- ۲ - فهرست مشترک نشریه های ادواری کتابخانه های ایران - بخش اول: کتابخانه های دانشگاهی و سازمانهای وابسته به وزارت فرهنگ و آموزش عالی، مرکز اسناد و مدارک علمی، اسفند ۱۳۶۱.

- ۳ - راهنمای انتشارات سازمانهای دولتی ایران (فهرست سالانه ۱۳۶۰)، مرکز اسناد و مدارک علمی، بهمن ۱۳۶۱.
- ۴ - فهرست انتشارات سازمانهای دولتی ایران موجود در مرکز اسناد و مدارک علمی، جلد ۴، جلد ۵، و جلد ۶، دی و بهمن ۱۳۶۱.
- ۵ - راهنمای مراکز اسناد و کتابخانه های تخصصی، اختصاصی، و دانشگاهی، مرکز اسناد و مدارک علمی، بهمن ۱۳۶۱.

نزدیک شدن انتخابات هیأت اجرایی

پنجمین مجمع عمومی سالانه انجمن باید بر طبق اساسنامه در مهرماه ۱۳۶۲ برگزار شود. چون دوره مسئولیت هیأت اجرایی فعلی انجمن در مهرماه امسال به پایان می رسد، یکی از مطالب مورد بحث در این جلسه، انتخاب هفت نفر اعضای هیأت اجرایی جدید برای مدت دو سال می باشد.

ورق بزنید



دو این شماره

- اخبار انجمن ... تغییر در ترکیب هیأت اجرایی - نشریات واصله به کتابخانه انجمن - نزدیک شدن انتخابات هیأت اجرایی - فرارسیدن هنگام تمدید عضویتها
- اخبار ایران ... سمینارهای وزارت صنایع - آکمیهای فعالیتهای کامپیوتری
- اخبار جهان ... پلاک گردن سربازها با ۸۰۰۰ مورد اطلاعات شخمی - ساخت گرده چرخان در هند
- محصولات جدید ... سریعترین آتیر کامپیوتر جهان - آدامک مصنوعی برای خنثی سازی بمب - توانا تپهای کامپیوتر شخمی آی بی ام ساخت ژاپن
- نواوریها ... ضبط عمودی برای گرده های مغناطیسی
- اعضا ... تغییرات در آمار اعضای انجمن - تمدید عضویتها
- نامه ها ... درباره شرایط ارسال مقالات
- سرگرمی ... پاسخ سرگرمی "سالهای دی مورگان"
- گزارش ... نظرات اعضا درباره انجمن
- گزارش کوتاهی درباره کامپیوتر دانشگاه کرمان
- مقاله ۱ ... طراحی نرم افزار - قسمت دوم (ترجمه بهروز پرهامی)
- مقاله ۲ ... چگونگی طراحی مجموعه دستورالعملهای کامپیوتر (ترجمه ابراهیم نقیب زاده مشایخ)
- مقاله ۳ ... نوعی مفتر زبان پیمیک با دقت محاسباتی زیاد (ترجمه بهروز پرهامی و میترا کاتوزیان)
- مقاله ۴ ... نقش مدلهای ریاضی و کامپیوتر در مطالعات منابع آب (مسعود خیام)



گزارش

نظرات اعضا درباره انجمن

در شماره قبلی ما هنامه، نتیجه نظرخواهی درباره اساسنامه جدید انجمن را اعلام نمودیم و وعده کردیم که در این شماره، نظرات و پیشنهاداتی را که اعضای گرامی انجمن در حاشیه آرای خود ارائه داده اند درج کنیم. در زیر، ابتدای نظراتی که به اساسنامه انجمن مربوط می شوند به ترتیب شماره ماده اساسنامه و سپس سایر نظرات و پیشنهادات به نظر شما می رسند.

۱ - پیشنهاد می شود که در ماده ۴، به جای "محاسن" کلمه "مزایا" به کار رود.

۲ - در ماده ۴، کلمه "خطرات" کمی سوال برانگیز است. بهتر نیست به جای آن کلمه "معایب" به کار رود؟

۳ - در ماده ۴، بند "ب"، عبارت "کوشش در جهت همکاری متقابل..." به "همکاری متقابل..." تغییر یابد.

۴ - در ماده ۴، قید شده است که انجمن می تواند در سایر نقاط کشور شعبه داشته باشد. به نظر بنده، با گسترش دادن این شعب در شهرستانها، کمکی بزرگ به پیشبرد معلومات اعضا شهرستانی خواهید نمود.

۵ - تغییرات و مواد جدید در جهت گویا تر شدن اساسنامه می باشد و به نظر اینجانب ماده ۱۹ به خوبی تدوین گردیده است.

۶ - با نحوه انتخاب افراد هیأت اجرایی برای بیش از دو دوره (ماده ۱۹، بند "ث") مخالفم.

۷ - در ماده ۱۹، بند "ث"، عبارت "حداکثر نیمی از" حذف شود.

۸ - ماده ۲۵، تطویل و توضیحات زیاد از حدی دارد که آن را ناخوشایند می کند. می توانست به صورت ساده "کلیه فعالیت های انجمن در چهارچوب قوانین و مقررات کشور صورت می گیرد" نوشته شود. اگر لازم باشد می شود در عبارت فوق بعد از "فعالیت های انجمن" اضافه کرد: "اعم از انتشاراتی و غیره".

۹ - در صورت انحلال انجمن، اموال آن باید قبل از اهداء، جهت تسویه حساب های موجود انجمن به کار روند و بدین منظور ممکن است به فروش برسند. لذا بهتر است ماده ۲۶ گویای این مطلب باشد.

۱۰ - به نظر اینجانب، اساسنامه انجمن از هر لحاظ بویائی و مطلوبیت لازم را دارا می باشد و تغییرات جزئی باعث تغییر کیفی و کمی انجمن نمی شود.

اعضا



تغییرات در آمار اعضای انجمن

افراد زیر در خردادماه ۱۳۶۲ به عضویت انجمن انفورما تیک ایران درآمده اند:

د	پروانه	اسماعیل زاده	۵۲۰۴
د	زهره	باطنی	۵۲۰۶
د	میترا	پورکمال	۵۲۰۲
د	امیرهایون (تا ۶۳)	جنگی تیموری	۵۱۸۹
ع	اسدالله	حسینی	۱۴۴۸
د	محمد	خالقی	۵۲۰۱
د	فریدبا	دانشگر	۵۱۹۸
د	زهره	دایی کنده	۵۱۹۶
ع	داریوش	رحیمی بهزاد	۱۴۴۷
د	علی اکبر	رفعی پور	۵۲۱۰
د	تورج (تا ۶۳)	رفیعی طاری	۵۱۹۴
د	سوسن	سعادت	۵۱۹۷
د	گیتی	شلیبری	۵۱۹۹
ع	عبدالحسین	صراف زاده	۱۴۴۹
د	علی (تا ۶۳)	طهرانی	۵۱۹۲
د	رومینا	عیسوی جنبزه	۵۲۰۵
د	علی اصغر	فاتی	۵۱۹۰
د	داود (تا ۶۳)	کرامتی	۵۱۹۳
د	سوریا عظم	کردبچه	۵۲۰۰
د	مریم	گل بهار حقیقی	۵۱۹۵
د	محمود (تا ۶۳)	مریخی	۵۲۰۷
د	آریتا	معتمد وزیری	۵۲۰۳
د	حسین	معماریان	۵۲۰۹
د	محمدرضا	مومن خانی	۵۱۹۱
ع	محمد	نخعی نژاد	۱۴۵۰
د	مهرداد	نوروزی	۵۲۰۸

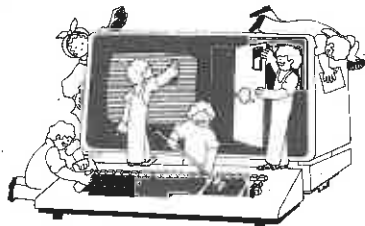
به این ترتیب آمار اعضای انجمن تا پایان خردادماه ۱۳۶۲ به قرار زیر است:

عادی	وابسته	دانشجویی	حقوقی	جمع کل
۳۰۲	۵۸	۱۳۰	۱	۴۹۱

تمدید عضویتها

طی خردادماه ۱۳۶۲، افراد زیر عضویت خود را در انجمن انفورما تیک ایران تمدید نموده اند:

ع	بهزاد اسیری	۱۳۶۲
ع	پرنیان	۱۱۷۶
ع	ترکمانی	۱۱۸۲
ع	سلامی	۱۰۶۲
ع	کریمیان	۱۲۲۳
ع	گلی	۱۰۷۸
ع	مجتهدی	۵۲۱۱
د	همایون	
ع	فرامرز	
ع	احمد (۶۳-۶۰)	
ع	عبدالله	
ع	حسام	
ع	سعید	
د	حمید	



دوستان و همکاران خود به تمدید عضویت، سیر صعودی تعداد اعضا و در نتیجه توفیق روزافزون انجمن را در ارائه خدمات علمی و تخصصی به جامعه کشورمان تضمین نمائید. ★

دنباله اخبار انجمن...

دستور جلسه مجمع عمومی و اساسی نایب های انتخابات هیأت اجرایی در شماره بعدی ما هنامه چاپ خواهد شد. همچنین ورقه اعلام رأی همراه با شماره بعدی برای کلیه اعضای دارای حق رأی ارسال خواهد شد تا این اعضا، در صورت عدم امکان شرکت در جلسه مجمع عمومی، بتوانند آرای خود را از طریق پست اعلام نمایند.

از اعضای عادی انجمن که مایل به فعالیت در جهت تحقق اهداف علمی و تخصصی انجمن انفورما تیک ایران هستند تقاضا می شود داوطلبی خود را برای عضویت در هیأت اجرایی، همراه با خلاصه ای از سوابق و تجربیات تخصصی خود برای چاپ در ما هنامه به رئیس انجمن اطلاع دهند. به خاطر داشته باشید که عضویت در هیأت اجرایی انجمن مستلزم دارا بودن وقت کافی و تمایل به انجام یک خدمت فرهنگی و اجتماعی بدون هیچ گونه چشمداشت مادی است.

فرارسیدن هنگام تمدید عضویتها

بدین وسیله به اعضای گرامی انجمن یادآوری می کنیم که زمان تمدید عضویتها نزدیک است. چون دوره یکساله عضویت در انجمن انفورما تیک ایران از مهرماه هر سال تا شهریورماه سال بعد است، کلیه افرادی که قبل از تیرماه ۱۳۶۲ به عضویت انجمن درآمده اند باید تا شهریورماه سال جاری عضویت خود را در انجمن تمدید نمایند.

افرادی که از تیرماه سال جاری به بعد عضو شده اند و نیز کسانی که حق عضویت دوساله را به صورت یکجا پرداخت کرده اند نیازی به تمدید عضویت ندارند.

سیر صعودی تعداد اعضا نشان می دهد که تا شهریورماه امسال، انجمن انفورما تیک ایران برای اولین بار در طول حیات خود، متجاوز از ۵۰۰ عضو خواهد داشت. امسال متأسفانه، انجمن تعدادی از این اعضا را به خاطر عدم تمدید به موقع عضویت از دست خواهد داد.

افرادی که عضویت خود را برای سال ۶۲ - ۱۳۶۱ تمدید نکرده اند، در شهریورماه امسال از لیست اعضای انجمن حذف خواهند شد. این قبیل افراد باید برای ادامه عضویت، حق عضویت دوساله ۶۳ - ۱۳۶۱ خود را (۱۵۰۰ + ۱۰۰۰ ریال برای اعضای عادی و وابسته و ۵۰۰ + ۳۰۰ ریال برای اعضای دانشجویی) حداکثر تا پایان شهریورماه ۱۳۶۲ بپردازند. سایر اعضا می توانند با پرداخت حق عضویت سال ۶۳ - ۱۳۶۲ (۱۵۰۰ ریال برای اعضای عادی و وابسته و ۵۰۰ ریال برای اعضای دانشجویی) عضویت خود را در انجمن تمدید نمایند.

شما عضو گرامی انجمن می توانید با پرداخت به موقع حق عضویت خود و با ترغیب

"اینترپات" درهم شکسته و معادلای فارسی "عملکرد چندبونا مه ای"، "پردازشگر" (یا "پردازنده")، و "وقفه" جای آنها را گرفته اند. ما سعی می کنیم تا با درج معادلای انگلیسی واژه های ناآشنای به کار رفته در هر شماره ما هنامه، خوانندگان جدید "گزارش کامپیوتر" را در تطبیق با این سبک جدید یاری دهیم. البته هنوز هم برخی اقوال آشنا با واژه های بیگانه و بیگانه با واژه های فارسی در مقابل ایس حرکت مقاومت می کنند. اما این حرکت حداکثر به اندازه یک نسل آموزشی دیگر یعنی تا هنگام فارغ التحصیل شدن دانشجویان جدید رشته کامپیوتر، دوام خواهد داشت. این متخصصان آینده به سادگی می پذیرند که "اینترپات" یک لغت عادی زبان انگلیسی است که در رشته کامپیوتر به تدریج بار تخصصی ویژه ای یافته است و همین بار تخصصی ویژه را می توان به لغت عادی "وقفه" در زبان فارسی منتقل نمود. در این باره گفتنی بسیار است و صفحات ما هنامه محدود★



نامه ها

درباره شرایط ارسال مقالات

آقای علیرضا ظاهری، یکی از اعضای دانشجویی انجمن انفورماتیک ایران، طی نامه مورخ ۱۳۶۲/۴/۶ خود یادآور شده اند که برای اکثر اعضای انجمن، امکان ماشین کردن مقالات وجود ندارد (یا حداقل بسیار اشکالاتی همراه است) و این شرط را مانعی در مقابل اعضاء برای ارسال مطالب دانسته اند. ایشان همچنین دو مطلب خبری، یکی درباره "سریعترین آتارکامپیوترهای جهان" و دیگری درباره "روش ضبط عمودی سونی" (که اولی در بخش محصولات جدید همین شماره درج شده است و راجع به دومی قبلاً مطلبی برای این شماره ما هنامه آماده شده بود) برای ما ارسال داشته اند.

ضمن تشکر از این عضو علاقمند، به اطلاع ایشان می رسانیم که اگر مقالات را با خط کاملاً خوانا و بارعایت فواصل کافی بین سطرها و حاشیه زیاد تهیه نمایند، از نظر ما اشکالی ندارد. شرط ماشین کردن مقالات تنها برای تسهیل و تسریع تهیه ما هنامه گذاشته شده است، چون در هر حال همه مطالب مجدداً در شکل و قالب خاصی برای ما هنامه ماشین می شوند. این تغییر را به زودی در شرایط ارسال مطالب منعکس خواهیم کرد.★



لازم به یادآوری است که برخی از تخفیرات اخیر اسانامه به پیشنهاد شورای عالی انفورماتیک کشور برای صدور اجازه ثبت انجمن صورت گرفته اند و هیأت اجرایی در تمام موارد با مضامین و عبارات پیشنهادی کاملاً موافق نبوده است.

(ب) در مورد نظری (سؤال؟) شماره ۱۴، باید گفت که سرمایه ابتدائی انجمن تنها از طریق حق عضویت های ۱۰۰۰ ریالی اعضاء اولیه تأمین گردید. در ماه های آغازین، که بنییه مالی انجمن نسبتاً ضعیف بود، کمک دانشگاه صنعتی شریف در زمینه چاپ ما هنامه گزارش کامپیوتر به صورت مجانی، ادامه فعالیت انجمن را مقدور ساخت. اما در حال حاضر، تمام هزینه های انجمن از طریق دریافت حق عضویتها و کمکهای مالی اعضاء نیز فروش نشریات به افراد و سازمانهای غیرعضو تأمین می شود. ما زاد درآمد انجمن نسبت به هزینه، در صورت وجود، به سرمایه انجمن اضافه می شود و موجبات گسترش فعالیتها و خدمات آن را فراهم می آورد. رسیدگی به حسابهای انجمن برطبق اسانامه برعهده هیأت اجرایی منتخب اعضاء (علی الخصوص خزانه دار و رئیس انجمن) است.

(پ) در مورد خط مثنی "گزارش کامپیوتر" (پیشنهادهای ۱۳ و ۱۹)، قبلاً هم توضیح داده ایم. "مسائل جاری و حائز ملاحظه کامپیوتری داخلی" باید توسط خود افراد شاغل در این مراکز (اعم از اعضاء انجمن انفورماتیک ایران یا افراد غیر عضو) و از جمله عضو عزیزی که این پیشنهاد را داده است مطرح شوند. ما نیز بیصبرانه در انتظار دریافت چنین مطالبی هستیم و مقالات دریافتی از این نوع را فوراً وبدون رعایت نوبت چاپ خواهیم کرد. در زمینه روان بودن مقالات نیز تلاش زیادی می کنیم. متأسفانه اغلب مقالات دریافتی از نظر دقت علمی و سبک نگارش بسیار ضعیف هستند. به ویرایش زیاد نیاز دارند. طبعاً با زیاد بودن حجم کار، کیفیت پائین می آید. اما شاید "روان نبودن مطالب برای برخی از خوانندگان، بیشتر به خاطر مشی مادر استفاده از واژه ها و عبارات فارسی باشد. گزارش کامپیوتر" اولین نشریه ای بود که از چهار سال قبل جرئت استفاده از واژه های فارسی را در زمینه ای از علم و تکنولوژی که پیش از آن به طور درستی تسلیم واژه های بیگانه شده بود به خود داد. اکنون با ازبیین رفتن مشکلات و مقادیمهای اولیه، این سبک تا حدود زیادی در میان متخصصان کامپیوتر و دست اندرکاران امور انفورماتیک جا افتاده است، به طوری که ما روز به روز شکایات و گلله های کمتری از جانب خوانندگان خود دریافت می کنیم. تقدس کاذب واژه هایی چون "مولتی پروگرامینگ"، "پرویسور"، و

۱۱ - پیشنهاد می شود که در اسانامه ماده ای به عنوان "تبلیغات" نیز گنجانده شود تا ماهیت انفورماتیک برای عموم مشخص شود و این علم در ایران امروز روبه زوال نرود.

۱۲ - به نظر اینجانب تقویت همکاری بین دو ارگان (انجمن و شورا) یکی از اساسی ترین نکاتی است که باید در اسانامه گنجانده شود.

۱۳ - بایستی با از طریق گنجانیدن در اسانامه و با قراردادن در خط مشیسی ما هنامه، مطالب این نشریه بیشتر دربرگیرنده مسائل جاری و حائز ملاحظه کامپیوتری داخلی باشد تا درج مطالبی چون آشنائی با زبانهای ناماء نوس و غیر مفید.

۱۴ - در این اسانامه مشخص نیست که چه کسانی در انجمن سرمایه گذاری نموده اند. هزینه ها از کجا تأمین می شوند، تکلیف سود و زیان انجمن چیست، و چگونه حسابهای انجمن رسیدگی خواهد شد. در هر حال زحمات شما قابل تقدیر است، متشکرم.

۱۵ - پیشنهاد می شود پس از تصویب اسانامه، آن را به صورت یک ورقه مجزا چاپ کنید و جهت کلیه اعضاء ارسال فرمائید. ۱۶ - جادارد که در زمینه ایجاد همکاری با موسسات خارج از کشور بیشتر فعالیت شود.

۱۷ - ضمن تشکر از زحمات هیأت اجرایی نظر این حقیر این است که نام انجمن نیز حتی المقدور متناسب با شرایط انقلاب اسلامی ایران تغییر یافته و حداقل کلمه "ایران" به عبارت "جمهوری اسلامی ایران" تغییر یابد.

۱۸ - نظری ندارم جز اینکه به ترتیب موجودیت انجمن حفظ شود و تا حد امکان سعی در توسعه فعالیت های علمی آن بشود. ۱۹ - لطفاً در صورت امکان مطالب علمی که از مجلات خارجی ترجمه می شود کمی روانتر تهیه شوند. چون بیشتر این مطالب قابل درک و قابل لمس نیستند (البته برای من). امیدوارم سایر خوانندگان گزارش کامپیوتر مشکل مرا نداشته باشند.

ضمن تشکر از افرادی که نظرات و پیشنهادهای فوق را ارائه داده اند و نیز اعضای که تعارفات، اظهار لطفها، و تشویقهای آنها را برای رعایت اختصار درج نکرده ایم، تذکر نکات زیر را لازم می دانیم:

(۲) نظرات و پیشنهادهای شماره ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸، ۹، ۱۱، ۱۲، و ۱۳ که به اعمال تغییرات مجدد در اسانامه مربوط می شوند در پرونده مخصوصی ثبت خواهند شد. از دیگر اعضاء گرامی انجمن که نظراتی در این زمینه دارند خواهشمندیم پیشنهادهای خود را برای ما بفرستند تا در نوبت بعدی که اعمال تغییرات در اسانامه لازم شود، این نظرات و پیشنهادهای مورد توجه قرار گیرند.

نوع "دک رایتر ۲"، یک کارت خوان، یک چاپگر ۳۰۰ سطر در دقیقه‌ای، یک تسهیم‌کننده ۲ ناهمگام ۱۶ خطی، و نرم‌افزار و ملزومات گوناگون (گرفته‌ها و نوارهای مغناطیسی، نوار چاپ، کارت‌های علامت‌زدنی، و غیره).

در حال حاضر بعضی از کارهای دانشگاه با این کامپیوتر انجام می‌شوند، دانشجویان نیز از آن استفاده می‌کنند. مشکلاتی که در رابطه با این کامپیوتر وجود دارند بیشتر در زمینه تعمیر و خرابی سخت‌افزار هستند و دور بودن کرمان از مرکز این مشکلات را تشدید می‌کند. تا مدتی تعمیر و نگهداری ماشین برعهده شرکت دیجیتال بود، اما بعد از آنکه گروهی از متخصصان این شرکت به استخدام شرکت ایزایران درآمدند، دانشگاه برای تعمیر و نگهداری با ایزایران قرارداد بست و هم اکنون نیز این قرارداد پابرجا است.

به علت ضعیف بودن دستگاه کارت خوان، و نیز کمبود دستگاه مگنته کارت و کارت مخصوص این ماشین، دانشجویان بیشتر با پایانه‌های سیستم کار می‌کنند و به همین جهت، محدودیت‌هایی برای استفاده از کامپیوتر ایجاد می‌گردد. قطع مداوم جریان برق و نداشتن ژنراتور از دیگر مشکلات ما است. اما به هر حال وجود همین کامپیوتر با تمام کمبودها و نارسا‌شده‌ها برای این منطقه کویری و به‌ویژه دانشگاه کرمان نعمتی است.

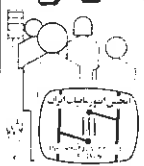
۱ - دک رایتر ۲ Decwriter II
۲ - تسهیم‌کننده Multiplexer
تهیه‌کننده: رحمت‌الله رئیسی اردکانی

مدل "اس‌ای ۸۰۱" را برای گرده‌های لیزران اینچی از همین ماه آغاز نماید. شرکت شوگارت ۲، که امتیاز ساخت این گرده‌ها را به شرکت هندی اعطاء کرده است، در نظر دارد که برای نفوذ بیشتر در بازار هندوستان، امتیاز ساخت محصولات دیگری را نیز به سازندگان هندی ارائه نماید. *

1. Sujata Data Products Private Ltd.
2. Shugart Associates

از "ویکی مایکرو بولتن"، ۱۶ آوریل ۱۹۸۳

گزارش کامپیوتر



ما هنامه انجمن انفورماتیک ایران زیر نظر کمیته انتشارات سرپرست کمیته انتشارات و سردبیر: دکتر بهروز پرهانی
نشانی: صندوق پستی ۲۵/۱۲۶
آریا شهر، تهران ۱۴

آخرین مهلت قبول مطلب و آگهی دو هفته پیش‌شده ماه انتشار است. نقل مطالب ما هنامه گزارش کامپیوتر با ذکر ماه خذ آزاد است.

گزارش



گزارش کوتاهی درباره کامپیوتر دانشگاه کرمان

دانشگاه کرمان در سال ۱۳۵۴ با بعضی از رشته‌های علوم تاسیس شد و تا قبل از انقلاب، فاقد هرگونه تجهیزات کامپیوتری بود. پیش از آن، بنیاد آرشام (نام رئیس سازمان امنیت وقت) شامل دانشسرای عالی فنی، انستیتوی تکنولوژی و غیره، درمکانی با ساختمان‌های نسبتاً مجهز، و از جمله ساختمانی به عنوان مرکز کامپیوتر، تاسیس شده و در سال ۱۳۵۷، یک کامپیوتر کوچک پی دی پی ۱۱/۷۰ به شرکت دیجیتال سفارش داده شده بود.

بعد از انقلاب، بنیاد آرشام به بنیاد جنیف نژاد و متعاقباً "پس از ادغام در دانشگاه کرمان، به دانشکده فنی دانشگاه کرمان تغییر نام یافت و کامپیوتر سفارش داده شده نیز در اوایل سال ۱۳۵۸ نصب گردید. این کامپیوتر، علاوه بر پردازنده مرکزی پی-دی پی ۱۱/۷۰، مشتمل بر تجهیزات جانبی زیر بود که جمعا "به قیمت تقریبی ۳۳۰ هزار دلار خریداری گردیدند: دو دستگاه گرده مغناطیسی ۶۷ میلیون بایتی، یک نوار چرخان برای نوارهای ۹ لبه ۸۰۰ و ۱۶۰۰ بیت در اینچ، سه پایانه نمایشگر، سه پایانه چاپگر از

اخبار ایران

سینماهای وزارت منابع

بنابر اطلاعیه مندرج در روزنامه کیهان مورخ ۱۳۶۲/۲/۱۲، وزارت منابع جهت بهره‌گیری از نظرات و تشکلات متخصصان در گروه‌های صنعتی مشابه برای بررسی مشکلات، امکانات توسعه کتی و کیفی، و آینده‌نگری، اقدام به تشکیل سینماهای بررسی صنایع در ۳۹ گروه مختلف نموده است. در زمینه صنایع برق و الکترونیک، دو سینما زیر برای اعضای گرامی انجمن و دیگر خوانندگان ما هنامه قابل توجهند:

صنایع ساخت نیمه‌هادیها، ۱۱ و ۱۲ آبان
کاربرد الکترونیک و ریزپردازنده‌ها در صنعت و آینده کشور
۲۱ و ۲۲ دی

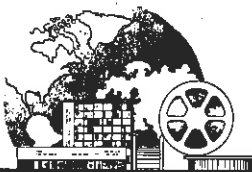
بنابر همین اطلاعیه، صاحبان، مدیران کارخانه‌ها، متخصصان، و علاقمندان می‌توانند حداکثر تا یک هفته قبل از تاریخ هر سینما، تمایل خود را به شرکت در جلسات سینما و یا ایراد سخنرانی در آن به وزارت صنایع اعلام نمایند.

آگهی‌های فعالیتهای کامپیوتری

موارد زیر به صورت آگهی در روزنامه کیهان درج شده بودند که جهت اطلاع خوانندگان گرامی خلاصه‌ای از آنها را می‌آوریم:

- شرکت ایران دیجیتال کامپیوتر کلاسهای را در زمینه آموزش نرم‌افزار دایر کرده است. نشانی: میدان آفریقا، ساختمان آلیتا لیا، طبقه ۶ (۶۲/۴/۲۵).
- مرکز تحقیقات مغایرات ایران ریزپردازنده کامپیوتر لاله خود را برای ساخت توسط واجدین شرایط فنی و تولیدی عرضه نموده است. تلفن تماس: ۶۳۸۰۶۱ (۶۲/۴/۲۶) *

اخبار جهان



پلاک کردن سربازها با ۸۰۰۰ مورد اطلاعات شخصی

در آینده نزدیک پلاک‌های براحتی قابل پاک شدن است و شناسایی فیزی که سربازان بگرم خود می‌بازند جای خود را به کارتهای پلاستیکی به همان اندازه سابق خواهند داد. بروی کارتهای جدید به کمک امواج الکترونیک ۸۰۰۰ مورد از مشخصات شخص از جمله نام، گروه خون، شماره ردیفه ایستی از حساسیت‌های سرباز، مذهب وی و غیره وجود دارد و به تنهایی یک پرونده پرسنلی کامل است. برای خواندن این کارت می‌توان آنرا در ماشین مخصوص قرار داد و براساسی اطلاعات الکترونیکی گدای مخصوص آنرا خواند. این اطلاعات در پلاک کردن سربازها با ۸۰۰۰ مورد اطلاعات شخصی

نقل از کیهان، ۲۲ تیرماه ۱۳۶۲

ساخت گرده چرخان توسط شرکت هندی

شرکت هندی "سوجاتا ویتا پروداکشن" در شهر بمبئی در نظر دارد که ساخت گرده چرخانهای

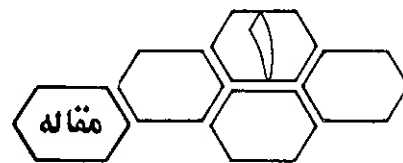


اهمیت نرم‌افزار

بررسی نشان داده است که خریداران کامپیوترهای شخصی در آمریکا بدواً به طور متوسط معادل ۶۰۸ دلار برای خرید نرم‌افزار صرف می‌کنند و طی سال اول استفاده از سیستم خود، معادل ۴۳۹ دلار دیگر نرم‌افزار می‌خرند. *

از "ویکی مایکرو بولتن"، ۱۶ آوریل ۱۹۸۳





طراحی نرم افزار

(قسمت دوم)

ترجمه آزاد و تلخیص: دکتر بهروز پرها می
دانشیار دانشگاه صنعتی شریف

پادداشت سردبیر: در زمینه طراحی نرم افزار گفتنی بسیار است و روزی نیست که کتاب یا مقاله جدیدی راجع به این موضوع چاپ نشود. در ماهنامه گزارش کامپیوتر نیز تاکنون مقالات چندین در این باره داشته ایم. در این مقاله توصیفی چند قسمتی، گذشته، حال، و آینده طراحی نرم افزار مورد بررسی قرار گرفته، خلاصه ای از سیر تحول این رشته، روشهای امروزی، مسائل و مشکلات موجود، و بالاخره امکانات و مسیر حرکت در دهه های که در پیش است ارائه می شود. در شماره گذشته بخشهای از مقاله را با عناوین زیر خواندیم:

- ۱ - مقدمه
 - ۲ - دیدگاه تاریخی
 - ۲/۱ - سرآغاز مسأله نرم افزار
 - ۲/۲ - محیط نسل دوم
 - ۲/۳ - سیستمهای نسل سوم
- در این شماره، پس از اتمام کار، در مجله "دیدگاه تاریخی"، مطالبی را درباره روشهای امروزی طراحی نرم افزار ارائه می دهیم.

۲/۴ - بحران اول نرم افزار

در اواخر دهه ۱۹۶۰ میلادی، براشیر شکستهای پی در پی سیستمها و طرحهای بزرگ کامپیوتری، رویاهای طلایی ناشی از خوشبینی و اغراق گوئی متخصصان و فروشندگان کامپیوتر یکی پس از دیگری فرو ریختند. گناه این شکستها نیز معمولاً برگردن نرم افزار بود. چون سخت افزارهای آن زمان عموماً در محدوده مشخصات و امکانات خود به خوبی کار می کردند.

در سال ۱۹۶۷ میلادی، گروهی از متخصصان نرم افزار که برای چاره جویی در زمینه مشکلات موجود گردهم آمده بودند، اصطلاح "مهندسی نرم افزار" را پدید آوردند. نسیا شاید بتوانند برخی از روشهای نظری و عملی دیگر رشته های مهندسی را در این زمینه به کار بندند. در واقع تشخیص این گروه این بود که مشکلات نرم افزار از نازکی و نوباشی این رشته ناشی می شوند و راه حل این مشکلات، جدی گرفتن مسأله و انجام مطالعات علمی و تجربی در زمینه طراحی نرم افزار است.

پیدایش مقاله "برنامه سازی ساخت یافته" و "طراحی ساخت یافته" در اوایل دهه ۱۹۷۰ میلادی را باید اولین شمره های این حرکت و دیور جدید به شمار آورد. در سال ۱۹۶۹ میلادی، دایکسترا ۱۷ یکی از مقالات کلیدی را در زمینه طراحی نرم افزار به رشته تحریر درآورد و طی آن تاءکید نمود که

طراحی نرم افزارهای بزرگ و پیچیده جز با به کارگیری روشهای ساخت یافته میسر نیست، "تجربید مسأله ۱۸" قبل از پرداختن به راه حل آن ضروری است، و بالاخره روش طراحی از بالا به پایین ۱۹ می تواند صحت برنامها را تضمین نماید.

تقریباً "در همان زمان، پاراناس^۱ مطالعاتی را در زمینه نحوه تعریف واحدهای مستقل نرم افزار و ترکیب آنها به صورت سیستمی با ساختار کلی مورد نظر آغاز کرد. وی یک تعریف خوب برای برنامه مورد نظر را تعریفی می دانست که کامل و دقیق بوده و در یک زبان قابل اجرا توسط ماشین بیان گسرد به طوری که بتوان از خود کامپیوتر برای تحقیق در کامل بودن و سازگاری اطلاعات ارائه شده سود جست.

اگرچه دایکسترا و پاراناس تنها کسانی نیستند که در زمینه حل مشکل طراحی نرم افزار کارهای مهمی انجام داده اند، اما مطالعات آنها را می توان مبنای خوبی برای ساخت بندی و تجربید مسأله، طراحی از بالا به پایین، و پیمانهای بودن^۱ نرم افزارها به شمار آورد.

بحران نرم افزار این شناخت را به وجود آورد که روش برنامه سازی "ذوقی" در مقابل به سیستمهای بزرگ نرم افزاری محکوم به شکست است و برای حصول ویژگیهای چون تحویل در موعد مقرر، قابلیت اطمینان، و سهولت کاربرد، باید از روشهای مهندسی استفاده نمود. همین شناخت بود که بعدها به پیدایش زمینه "طراحی نرم افزار" به عنوان بخشی از رشته "مهندسی نرم افزار" منجر گردید (۲).

۲/۵ - رشد و نفوذ سیستمی

در اوایل دهه ۱۹۷۰ میلادی، رشد چشمگیر امکانات سخت افزاری با استفاده از تکنولوژی مدارهای مجتمع الکترونیکی آغاز گردید. این رشد با تمرکز امکانات محاسباتی به منظور بهره گیری از مزایای اقتصادی عملکرد در مقیاس بزرگ همراه گردید. کارآئی سیستم با میزان اشتغال پردازنده مرکزی یا هزینه صرف شده در آرای هسیر واحد وقت پردازنده مترادف بود. اندازه کامپیوترها بزرگتر شد: آی بی ام از سری ۴۶۰ به سری ۴۷۰ (مدلهای ۱۵۸ و ۱۶۸) روی آورد و سی دی سی سیستم ۷۶۰۰ خود را جایگزین سیستم ۶۶۰۰ نمود.

کامپیوترهای فوق العاده بزرگ معمولاً برای کاربردهای تجاری و علمی به کار می رفتند. برنامه سازان آشنائی کاملی با سیستم مورد استفاده و شناخت نسبتاً خوبی از مسائلی که باید حل می شدند داشتند و این مسائل را به صورت تبدیلات ساده ای از داده های ورودی به داده های خروجی در نظر می گرفتند.

اما به صورت هم زمان با این تمرکز گراشی و کاربزد کامپیوترهای فوق العاده بزرگ، دو جریان مخالف نیز تدریجاً خود را نشان می دادند. جریان اول، یعنی توسعه کاربزد کامپیوترهای کوچک، از اواسط دهه ۱۹۶۰ میلادی با ظهور پی دی پی-۸ آغاز گردید. کامپیوترهای کوچک فوق العاده ساده بودند، کلمات کوتاه و دستورالعملهای محدودی داشتند، فضای نشانی دهی آنها نسبتاً کوچک بود، و عملیات ورودی و خروجی را به صورت مستقیم (بدون استفاده از میانگیرها یا مجراهای ورودی و خروجی) انجام می دادند. اما ارزانی نسبی کامپیوترهای کوچک این امکان را به استفاده کننده می داد که بدون نگرانی از پائین بودن میزان اشتغال پردازنده، مرکزی، یک کامپیوتر را برای انجام یک کار یا دسته ای از کارهای بخصوصی اختصاص دهد.

جریان دوم به کاربرد تکنولوژی مدارهای مجتمع الکترونیکی در مقیاس بزرگ برای ساخت ریزپردازنده ها مربوط می شد. در بدو امر، امکانات این ریزپردازنده ها حتی از کامپیوترهای کوچک نیز کمتر بود. اما پائین بودن هزینه این واحدها سبب شد که کاربرد آنها به سرعت گسترش یابد و مفهوم "کامپیوتر شخصی" به تدریج به واقعیت بپیوندد.

از اواخر دهه ۱۹۶۰ میلادی به بعد، نرم افزارهای پشتیبان برای کامپیوترهای بزرگ گسترش چشمگیری یافتند. کاربزد زبانهای سطح بالا روبه فزونی گذاشت. زبانها بترن و کوپول به صورت استاندارد درآمدند. زبان الگول هنوز به کار می رفت و بخصوص در اروپا طرفداران زیادی داشت. آی بی ام سعی می کرد که زبان پی ال وان را جای بیا نندازد و زبان لی پی ال نیز به تدریج جای خود را بازمی کرد. بازار نرم افزار کامپیوتر ملو از کامپایلرها، سیستمهای عامل، سیستمهای مدیریت بانکهای اطلاعاتی، و برنامه های کمکی و پشتیبان بود و مفاهیمی چون حافظه مجازی برای تسهیل کار برنامه سازان به کار گرفته می شدند. اما هنوز هم وقت پردازنده مرکزی از هر چیز دیگری با ارزشتر بود و مبنای محاسبه هزینه اجرای نرم افزارها قرار می گرفت.

محیط برنامه سازی برای کامپیوترهای کوچک و ریز کامپیوترها کاملاً با آنچه که در مورد کامپیوترهای بزرگ گفتیم متفاوت بود (و هنوز هم هست). در آغاز، این محیط به محیط نرم افزاری کامپیوترهای نسل دوم شباهت داشت: نرم افزارهای پشتیبان فوق العاده ضعیف بودند و برنامه سازان با ارقام هشت هشتی با شانزده شانزدهی زبان ریز کامپیوترها سروکار داشتند (وضعیت در مورد کامپیوترهای کوچک کمی بهتر بود).

ورق بزنید

اما به تدریج با گسترش بازار، تعداد ریز-کامپیوترهای مشابه به اندازه‌ای رسید که طراحی و بازاریابی نرم‌افزار برای آنها صرف می‌گردد. با این حال هنوز هم سیستم‌هایی وجود داشتند که به تعداد کم و برای موارد بسیار خاصی (مثلاً: مخابرات نظامی) تولید می‌شدند. نرم‌افزارهای این قبیل سیستمها، برای اجتناب از هزینه فوق‌العاده زیادی طراحی و نگهداری نرم‌افزارهای منحصر به فرد، بسیار ساده و ابتدائی بودند.

۲/۶ - بحران دوم نرم‌افزار

علیرغم مطالعات جدی و پیگیری که بر اثر بحران اول نرم‌افزار در اواخر دهه ۱۹۶۰ میلادی آغاز شدند، هنوز هم نرم‌افزارها: (۱) دیرتر از موعد مقرر آماده می‌شدند. (۲) فوق‌العاده گران‌از آب درمی‌آمدند. (۳) قابلیت اطمینان کافی نداشتند. (۴) انتظارات استفاده‌کننده را بر نمی‌آوردند.

اما پاسخ متخصصان به این بحران نسبت به دفعه قبل متفاوت بود.

به جای بررسی مشکل به صورت یک مسأله نظری، جنبه‌های عملی آن و بخصوص موضوع مدیریت طرحهای ایجاد نرم‌افزارها، مورد تأکید بیشتر قرار گرفتند. دیدگاه مهندسی بر تمام مراحل ایجاد نرم‌افزارها حاکم گردید^{۲۲}. کتابهایی در زمینه مدیریت طرحهای نرم‌افزاری به رشته تحریر درآمدند^{۲۳، ۲۴}. نتایج مطالعات و بسط در باره روشهای غیرمتعارف مدیریت طرحها منتشر شدند^{۲۵}. برنامه‌سازی ساختاریافته ابتدا به عنوان روشی برای افزایش بازدهی و متعاقباً "برای مدیریت گروههای برنامه‌سازی کاملاً" جا افتاد^{۲۶، ۲۷}. این تأکید بر مدیریت طرحهای نرم‌افزاری، مفهوم "چرخه زندگی"^{۲۸} نرم‌افزار را به وجود آورد. اگرچه بحث در این زمینه خارج از حیطه موضوع مقاله (یعنی طراحی نرم‌افزار) است اما برای آنکه بتوانیم نقش مرحله طراحی و اهمیت آن را در چرخه زندگی یک سیستم نرم‌افزاری ببینیم، مراحل پنجگانه این چرخه را به اختصار مرور می‌کنیم:

تحلیل - این مرحله با تعریف مشخصات سیستم آغاز شده و با بررسی و پذیرش طرح اولیه، سیستم خاتمه می‌یابد. در این مرحله، وظایف و ویژگیهای کلی و نیز کارآئی برنامه‌های مورد نیاز مشخص می‌شوند. این مشخصات اولیه مبنای بررسیها و عملیات بعدی در تمام مدت چرخه زندگی سیستم نرم‌افزاری قرار می‌گیرند.

طراحی - در این مرحله، مشخصات اولیه سیستم به یک روش طراحی، شامل مدلهای ریاضی لازم و نمودارهای گردش کلی، تبدیل می‌شوند. روابط بین واحدهای مختلف نرم‌افزاری مشخص شده و پیش‌نویس مشخصات

برنامه‌ها تهیه می‌گردد.

برنامه‌سازی: در این مرحله، تمام برنامه‌های لازم ایجاد و اشکال زدائی می‌شوند. سپس مولفه‌های ایجاد شده به صورت گروهی مورد آزمایش قرار گرفته و تدریجاً بارفع اشکالات، سیستم کامل را به وجود می‌آورند. آزمایش اولیه، مستند سازی، و تعیین رویه، آزمایش‌های سیستم این مرحله را کامل می‌کند.

آزمایش و تلفیق - در این مرحله، نرم‌افزار ایجاد شده با سیستمهای سخت-افزاری لازم تلفیق شده و آزمایشهای دقیقی برای تعیین آنکه آیا سیستم کلی دارای مشخصات مورد نظر (تعیین شده در مرحله تحلیل) می‌باشد به عمل می‌آیند. در پایان این مرحله، عملکرد نرم‌افزار برای مشتری به نمایش در می‌آید و دوره ایجاد خاتمه می‌یابد.

اجرا و پشتیبانی - در این مرحله، عملکرد سیستم تحت شرایط واقعی کار بررسی می‌شود و تغییرات و اضافات لازم در آن اعمال می‌گردند. این تغییرات و اضافات تنها برای رفع اشکالات باقی مانده در طراحی اولیه نرم‌افزار مورد نیازند و یا آنکه به منظور وفق دادن سیستم با شرایط و محیط‌های جدید یا افزایش کارآئی آن ضرورت پیدا می‌کنند.

بررسی چرخه زندگی نرم‌افزار نه تنها سیب‌بند که شما نیز لازم بین طراحی نرم‌افزار و برنامه‌سازی به وجود آید، بلکه پیدایش بسیاری از روشهای طراحی امروزی را موجب گردید. بنا بر این، بحث مختصر فوق، زمینه مناسبی را برای بررسی روشهای متداول طراحی نرم‌افزار، با تأکید بر مرحله طراحی و رابطه آن با مرحله تحلیل، فراهم می‌کند.

۳ - روشهای امروزی

امروز طراحی نرم‌افزار با روشهای متعددی صورت می‌گیرد. تمام این روشها بر اصل "طراحی از بالا به پایین" مبتنی هستند. برخلاف روش سنتی طراحی از پایین به بالا، که در آن عناصر و اجزای سیستم مستقلاً طراحی شده و سپس در مرحله تلفیق سیستم برای اولین بار در کنار هم گذاشته می‌شوند، در طراحی از بالا به پایین، پیش از مشخص شدن ساخت کلی سیستم، هیچ اقدامی در زمینه تعریف جزئیات و برنامه نویسی صورت نمی‌گیرد. به این ترتیب، از بروز مشکلات غیرمنتظره در مرحله تلفیق سیستم (که علل اصلی پیدایش "وصله‌های" ناجور و درهم ریختن ساخت منطقی سیستم را تشکیل می‌دهند) جلوگیری می‌شود. در روش طراحی مرحله‌ای از بالا به پایین، که امروز به عنوان یک روش مهندسی از محبوبیت

زیادی برخوردار است، مراحل طراحی از تعاریف سطح بالای کنترل و داده‌ها آغاز شده و با مشخص نمودن واحدهای نرم‌افزاری ساختمان داده‌ها ختم می‌شوند. در این روش، تأکید بر پالایش گام به گام^{۲۹} طرح، از بالاترین سطوح ساختی تا پایین‌ترین سطوح مربوط به واحدها و عملیات ساده، قرار دارد^{۳۰}. این روش منظم و منطقی برای طراحی، ایجاد تغییرات و تلفیق اجزای کلی طرح را پیش از اقدام به برنامه نویسی ممکن می‌سازد.

یکی از مهمترین جنبه‌های روش طراحی از بالا به پایین، اصرار آن بر وجود تعاریف دقیقی از نیازها است. بنابراین، توصیف مختصری از محیط طراحی و نحوه مشخص نمودن نیازها برای درک روشهای امروزی طراحی نرم‌افزار ضروری به نظر می‌رسد.

۳/۱ - محیط طراحی

اغلب روشهای متداول برای هدایت مراحل طراحی نرم‌افزار بر اساس تعاریف دقیقی از نیازها، بر نظریه‌های عمومی مهندسی سیستم و روشهای تحلیلی مبتنی هستند. سیستمی که در اینجا مورد نظر ما است از سخت‌افزار، نرم‌افزار، و سایرین ارشباطی، و نیز گنشها و واکنشهای استفاده‌کننده تشکیل می‌شود. برخی از روشهای مهندسی سیستم که برای ساخت بندی مسأله مورد استفاده قرار می‌گیرند در زیر بررسی می‌شوند.

نمودارهای گردش وظایف^{۳۱}: این نمودارها بیانگر فرآوردهای سطح بالا در سیستم هستند. فرآوردهای لازم در ساخت کلی سیستم به طور ذهنی استخراج شده و با واسطهای^{۳۲} مناسبی با یکدیگر ارتباط می‌یابند. نمودار حاصل هم به عنوان یک پیکربندی^{۳۳} ابتدائی برای سیستم و هم به عنوان راهنمایی برای تقسیم وظایف و ایجاد هماهنگی بین چند تحلیل گر مفید است.

نمودارهای دنباله‌ای وظایف^{۳۵}: این نمودارها مشابه نمودارهای قبلی هستند اما نسبت به آنها از کلیت بیشتری برخوردارند چون به جای استفاده از روابط ایستا بین فرآوردها، وظایف سیستم را بر حسب دنباله‌های زمانی بیان می‌کنند.

نمودارهای دنباله‌ای گنشه^{۳۶}: این نمودارها وسیله‌ای برای تعریف نمودن و تخصیص دادن گنشه، تصمیمات، و فرآوردهای لازم برای ترجمه نیازهای سیستم به مشخصات نرم‌افزارها به شمار می‌روند. تحلیل گر دنباله‌های کاملاً اتمام گنشه‌های مربوط به کاربردهای مختلف (ثاملاً اپراتور، زیرسیستم، تجهیزات، و رابطهای تسرم-افزاری) ایجاد می‌کند. برای مشخص شدن رابطه کامل زیرسیستم با محیط عملکرد آن، فهرستی از تمام اپراتورها، تجهیزات و

درباره کنترل نیست و ترتیب زمانی عملیات را نیز نشان نمی دهد. بلکه فقط مراحل انجام کار را برای تبدیل اطلاعات ورودی به اطلاعات خروجی دربر دارد. این نوع نمودار به عنوان تقریب اولیه ای برای نشان دادن عملکرد سیستم و داده های که با آنها سروکار دارد قابل استفاده است (نمودار ۲). نمودار جریان داده ها و نمودار ساختی به یکدیگر قابل تبدیلند.

نمودار ساختی: هدف نمودار ساختی نشان دادن سازمان سیستم به صورت سلسله مراتبی است، به طوری که روابط بین واحدها و نحوه کنترل به خوبی مشخص باشد. در تبدیل نمودار جریان داده ها به نمودار ساختی، تبدیلات اولی به صورت واحدهای موجود در دومی جلوه گر می شوند. تجزیه واحدها به صورتهای گوناگونی که بعداً مورد بحث قرار خواهند گرفت امکان پذیر است. زیرا واحدها، که در سطحی پایین تر از واحدها قرار دارند، اطلاعات را از واحدها دریافت می کنند و نتایج را به آنها می فرستند. علائم تصویری ویژه ای نیز برای نشان دادن تکرار، عوامل تصمیم گیری، و غیره موجودند (نمودار ۳). این واقعیت که نمودار ساختی باید دارای خصیصه های معینی باشد برای ارزیابی وضعیت طرح به کار می رود. نمودارهایی که از این الگوهای معین تبعیت نکنند عموماً "به طرحهای ناقص، پیچیده، یا نادرست منجر می شوند.

توصیف جدولی: داده های سیستم به صورت جداولی، که نام گروه داده ها را به خود می گیرند، توصیف می شوند. در ابتدای کار داده ها به صورت ذهنی، به شکل جداول وابسته به نمودار ساختی، تعریف می شوند. با تجزیه واحدهای پردازش در سیستم، محتوی جداول با توجه به داده های وارد شده به آنها و داده های خارج شده از آنها مورد شناسائی قرار می گیرد.

۳/۴ - معنی و پویایی

تصور ذهنی ما از آنچه که یک سیستم طرح شده انجام می دهد، به معانی واژه های مورد استفاده در توصیف آن وابسته است. این قبیل واژه ها برای مؤثر بودن باید دارای خصوصیات زیر باشند:

- ۱ - واژه ها باید توصیفی باشند و معانی مورد نظر را به تمام افراد دست اندر کار طراحی برسانند. طی مراحل طراحی، نهایتاً باید قراردادهای را در نام گذاری داده ها و واحدها رعایت کرد تا این نامها با خصوصیات و محدودیتهای زبان برنامه سازی مورد استفاده تطبیق نمایند.
- ۲ - واژه ها باید دقیق و فاقد ابهام معنایی باشند. علائم اختصاری و دیگر نامهایی که احتمال سوء تعبیر در موردشان وجود دارد باید در یک واژه نامه دقیقاً ورق بزنید

۳/۲ - روشها و فنون

روشهایی که امروز در طراحی نرم افزار به کار می روند متعدّدند. هر یک از این روشها دارای ویژگیها و عناصر خاص خود است. قابل تصور است که روزی تمام این عناصر مختلف تلفیق شده و یک روش واحد و جامع را برای طراحی نرم افزار به وجود آورد. اما از طرف دیگر، این امکان نیز وجود دارد که روزی عدم توانائی یک روش واحد برای حل تمام انواع مسائل به ثبوت برسد. مثالهای زیر روشن می کنند که روشهای مختلف طراحی نرم افزار از محیطهای گوناگون ناشی شده و دیدگاههای متفاوتی را مورد تأکید قرار می دهند.

روش طراحی ساخت یافته با تعقیب جریان داده ها در درون سیستم آغاز می شود، اما نهایتاً "بر شکستن و تفکیک واحدهای نرم افزار تمرکز می باید ۲۸". جکسون ۴۰ به داده ها به عنوان نیروی محرکه طراحی می نگرد و برنامه را تنها وسیله ای برای تبدیل داده های ورودی به داده های خروجی می داند. وارنیه ۴۱ نیز داده ها را مورد تأکید قرار می دهد، اما روش منظم تری را برای طراحی در نظر می گیرد. روش تحلیل و طراحی ساخت یافته (یا ساخت ۴۲) بیشتر بر نمایش تصویری طرح و شکستن و تفکیک نیازها به صورت سیستماتیک تأکید دارد ۴۳.

در تمام این روشها، الگوی مشترکی به چشم می خورد. اول اینکه نحوه نمایش در یک روش خوب طراحی نرم افزار باید به فوق العاده واضح باشد و این خاصیت عموماً "توسط نمایش تصویری تأمین می شود. دوم اینکه در به کار بردن واژه ها و اصطلاحات حساسیت زیادی وجود دارد. ثالثاً مفاهیم و عناصر طراحی فاقد هر نوع ابهام باشند. سوم اینکه با توجه به پویا بودن فرایند طراحی، باید نوعی تکرار و امکان بازنگری و انتخاب مجدد در یک روش خوب طراحی نهفته باشد. چهارم اینکه چون طراحی از مرحله مفاهیم تجریدی آغاز شده و به نمایش یک راه حل مشخص ختم می گردد، روش طراحی باید دربرگیرنده سطوح مختلف نگاه کردن به مسأله باشد. بالاخره، پنجم اینکه به خاطر محدودیتهای زمانی، پرسنلی، و سخت افزاری، نوعی بهینه سازی بر اساس این محدودیتهای مورد نیاز است.

۳/۳ - نمایش اجزاء و مراحل کار

سه نوع ابزار اصلی در نمایش تصویری برای طراحی نرم افزار عبارتند از: نمودار جریان داده ها ۴۴، نمودار ساختی ۴۵، و توصیف جدولی.

نمودار جریان داده ها: هدف نمودار جریان داده ها نشان دادن تبدیلاتی است که حین عبور داده ها از سیستم بر آنها اعمال می شوند. این نمودار دارای اطلاعاتی

زیرسیستمهایی که باید بایک زیرسیستم بخصوص رابطه برقرار کنند تهیه می شود. نمودارهای دنباله ای عملکردی ۴۷: این نوع نمودار حرکت داده ها را طی مراحل گردآوری، پردازش، و بالاخره ارائه نتایج خروجی در سیستم نشان می دهد. مسیر حرکت هر نوع داده و فعل و انفعالات انجام شده بر روی آن در ارتباط با سیستم و داده های دیگر به دقت مشخص می شود. پایانه های اپراتورها (همراه با مراحل عملکرد هر اپراتور) و نیز پرونده های سیستم در این نمودارها به چشم می خورند.

پس از ساخت بندی مسأله، روشهای مختلف تقسیم وظایف بین سخت افزار، نرم افزار، و گنشهای انسانی مورد بررسی و مقایسه قرار می گیرند. یکی از راههای مقایسه کمی این راهحلهای مختلف استفاده از شبیه سازی است. انتخابتهای از بین راهحلهای مختلف با توجه به عوامل محدودیت هائی چون زمانی پاسخ ۴۸، تجهیزات سخت افزار موجود، بودجه طرح، مهارتها و نیروی انسانی موجود، و بالاخره برنامه زمان بندی شده طرح صورت می گیرد.

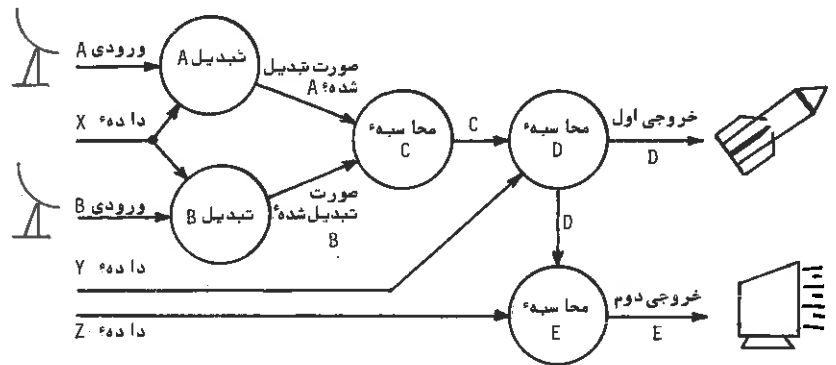
یک وسیله مفید برای ارزیابی راه حلها، مختلف، "ماتریس وزنی" است. ابتدا فهرستی از ضوابط مهم تهیه می شود و به هر ضابطه وزنی اختصاص می یابد. سپس میزانی برای ارزیابی هر ضابطه تعیین می شود و ارزش عددی هر راه حل در ارتباط با آن ضابطه به دست می آید. این ارزشهای عددی در وزنهای مربوطه ضرب شده، امتیاز کلّی هر راه حل محاسبه می گردد. راهحلی که بالاترین امتیاز را بیاورد انتخاب می شود. حصول اطمینان از مطلوبیت راه حل انتخاب شده، از طریق "تحلیل حساسیت" ۴۹ (تعیین اینکه تغییر دادن هر یک از عوامل تاجه حد بر نتیجه نهائی اثر می گذارد) امکان پذیر است.

راه حل نهائی انتخاب شده هم با روش شبیه سازی (برای به دست آوردن معیارهای کمی کارائی) و هم با نقد و بررسی توسط متخصصان (برای ارزیابی کیفی میزان تطابق راه حل با نیازهای اولیه) واری و تأیید می شود.

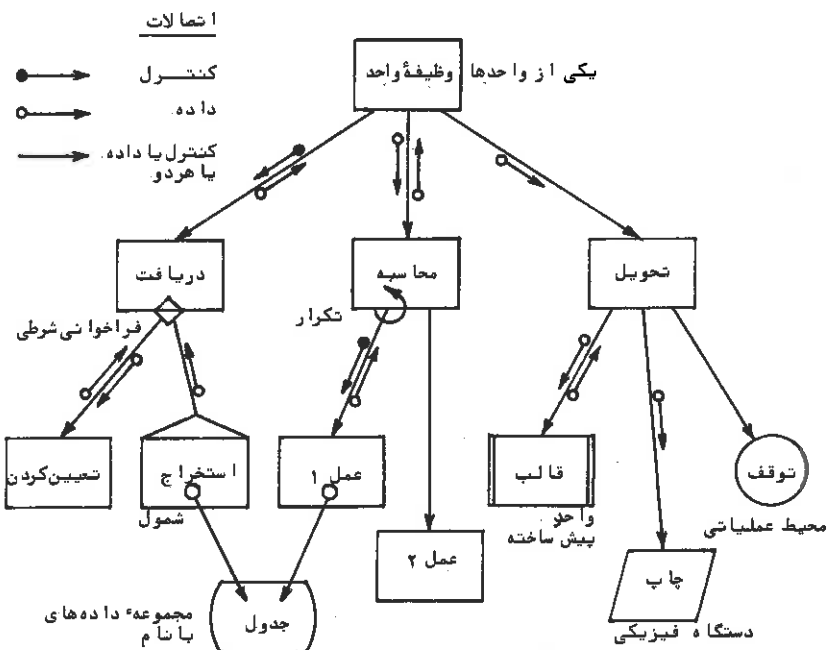
وظایف تخصیص یافته به نرم افزار توسط مدارک "بیان کننده نیازها" و "بیان کننده کارائی برنامه ها" مستند می شوند. در این مدارک، مشخصات داده های ورودی سیستم، نوع پردازشهای انجام شده، و خروجیهای مورد انتظار آن ارائه می شوند. نحوه ارائه این مشخصات به صورتی است که نتایج آنها طراح سیستم را در انجام وظایفش یاری می دهد، بلکه اطلاعات جامعی در بسار عملکرد و کارائی مورد انتظار سیستم در اختیار استفاده کنندگان نهائی و نیز متخصصانی که باید آزمونهای پذیرش نرم افزار را طراحی نمایند می گذارد.

یادداشتها و مراجع (قسمت دوم)

- ۱۷ - این ماه خن مقاله اصلی توسط ماخذ معادل زیر، که دستیابی به آن ساده‌تر است، جایگزین شده است:
Dijkstra, E.W., "Notes on Structured Programming," in *Structured Programming*, O.-J. Dahl et al, Academic Press, 1972.
- ۱۸ - تجربیدمساهله
Problem abstraction
- ۱۹ - از بالا به پایین
Top-down
20. Parnas, D.L., "ON the Criteria to Be Used in Decomposing Systems into Modules," *Communications of the ACM*, Dec. 1972.
21. Wasserman, A.I., "Software Engineering: The Turning Point," *Computer*, Vol. 11, No. 9, pp. 30-41, Sep. 1978.
22. Boehm, B.W., "Software and Its Impact: A Quantitative Assessment," *Datamation*, Vol. 19, No.3, pp. 48-59, May 1973.
23. Brooks, F.P., Jr., *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.
24. Weinberg, G.M., *Psychology of Computer Programming*, Van Nostrand, 1971.
25. Mills, H.D. and F.T. Baker, "Chief Programmer Teams," *Datamation*, Vol. 19, No. 12, Dec. 1973.
26. Stephens, W.P., G.J. Meyers, and L.L. Constantine, "Structured Design," *IBM Systems Journal*, Vol. 13, No. 2, pp. 115-139.
27. Yourdon, E. and L.L. Constantine, *Structured Design*, Yourdon, Inc., New York, 1975.
- ۲۸ - چرخه زندگی
Life cycle
- ۲۹ - پالایش گام به گام
Stepwise refinement
30. Wirth, N., *Systematic Programming*, Prentice-Hall, 1975.
- ۳۱ - نمودار گردش وظایف
Functional flow diagram
- ۳۲ - فراروند
Process
- ۳۳ - واسطه
Interface
- ۳۴ - پیگیری
Configuration
- ۳۵ - نمودار دنباله‌ای وظایف
Functional sequence diagram
- ۳۶ - نمودار دنباله‌ای کنشها
Action sequence diagram
- ۳۷ - نمودار دنباله‌ای عملکردی
Operational sequence diagram
- ۳۸ - زمان پاسخ
Response time
- ۳۹ - تحلیل حساسیت
Sensitivity analysis
40. Jackson, M.A., *Principles of Program Design*, Academic Press, 1975.
41. Warnier, J.D., *Logical Construction of Programs*, Van Nostrand, 1976.
42. SADT = Structured Analysis and Design Technique
43. Ross, D.T. and K.E. Schoman, Jr., "Structured Analysis for Requirement Definition," *IEEE Transactions on Software Eng.*, Jan. 1977.
- ۴۴ - نمودار جریان داده‌ها
Data-flow chart
- ۴۵ - نمودار ساختی
Structure chart



نمودار ۲ - مثالی از یک نمودار جریان داده‌ها



نمودار ۳ - مثالی از یک نمودار ساختی

یکی از جنبه‌های بنیادین بودن فراروند طراحی این است که هرگاه آن بر گامهای قبلی اثر می‌گذارد. با پیشرفت مراحل تحلیل یک واحد بخصوص، اثر تفکیک آن به اجزای کوچکتر باید در رابطه با تصمیمات فرضیات و نتیجه‌گیریهای قبلی در مورد سیستم بررسی شود. هدف فراروند طراحی، تعریف و پالایش پیوسته اطلاعات مربوط به سیستم همراه با تصمیم‌گیری درباره هریک از جزئیات آن است.

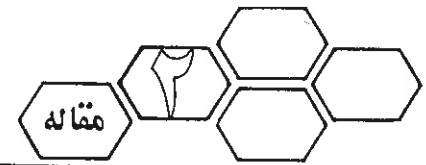
در مراحل ابتدائی طراحی، این آثار بسیار وسیع هستند و تقریباً "در تمام گامهای قبلی منعکس می‌شوند. در نتیجه، ساخت کلی سیستم به‌طور چشمگیری نوسان می‌کند. اما با پیشرفت طراحی، دامنه این آثار محدودتر می‌شود تا آنکه بالاخره در مراحل نهائی، چنین تصمیماتی تنها دارای آثار موضعی خواهند بود. ★

(ادامه دارد)

تعریف شوند. واژه‌ها باید به‌طور کامل و دقیق، وظایف واحدها و نتایج خروجی حاصل از آنها را توصیف کنند.

۳ - واژه‌ها باید کامل باشند. بسا مشخص شدن جزئیات عملیات داخلی سیستم، واژه‌ها به‌تدریج دقیقتر و قطعی‌تر می‌شوند تا آنکه نهایتاً "نام انتخاب شده وضعیت کامل واحد و داده‌های مربوطه را بیان نماید. تنها با ارزیابی درجه کامل بودن یک نام می‌توان از دقت آن اطمینان حاصل کرد.

۴ - واژه‌ها باید از نظر دستوری ساده باشند. واژه‌ها و ساختارهای ترکیبی کسبه نشان دهنده انجام چند کار هستند نباید به‌کار روند، چون این امر بایکی از اصول طراحی خوب که همانا تفکیک سیستم به واحدهای ساده‌ای است که یک وظیفه کوچک و معین را انجام می‌دهند مغایرت دارد.



چگونگی طراحی مجموعه دستورالعمل‌های کامپیوتر

ترجمه: ا. ابراهیم نقیبزاده؛ مشایخ

با دداشت سردبیر: در خلال ۲۵ سال گذشته، کامپیوترها و تکنولوژی سازماندهی، معماری و پیاده‌کردن آنها تکامل قابل ملاحظه‌ای یافته است. پاره‌ای از این دگرگونیها، که به کمک تکنولوژی مجتمع سازی در مقیاس خیلی بزرگ صورت گرفته‌اند، در واقع جهش‌های انقلابی بوده‌اند. اما در طول این مدت، مجموعه دستورالعمل‌های کامپیوتر تغییر مهمی نکرده است. دلیل این امر این است که هنوز به رشد لازم جهت بیان اهداف مجموعه دستورالعمل‌های کامپیوتر و سپس پیگیری دنباله‌ای از گام‌های اصولی و سیستماتیک برای طراحی و پیاده‌کردن چنین مجموعه‌ای نائل نگشته‌ایم. در این مقاله، چگونگی طراحی مجموعه دستورالعمل‌های کامپیوتر مورد بررسی قرار گرفته و معیارهای برای انتخاب و پیاده‌کردن آنها ارائه شده است.

۱- مقدمه

در زمینه ارائه سازماندهیها و معماریهای مختلف در کامپیوترهای جدید، کوششهای بسیاری شده است. اما مطالعه مجموعه دستورالعمل‌های کامپیوترهای جدید نشان می‌دهد که دگرگونی چشمگیری در این زمینه حاصل نشده است. معماران و متخصصان کامپیوتر ممکن است از بابت پیشرفتهای ظریفی که در کامپیوترهای جدید دیده می‌شوند به خود بیالند، ولی استفاده کنندگان معمولی حتی متوجه چنین دگرگونیهای نیز نمی‌شوند.

برای بسیاری از استفاده کنندگان، حتی اگر به زبان اسمبلی هم برنامه ننویسند، احتمالاً مجموعه دستورالعملها اولین محل برخورد با کامپیوتر است. همچنین، مجموعه دستورالعملها تاءثیر بزرگی در کارآئی سیستم عامل و نیز برنامه‌های مترجم و سایر برنامه‌هایی که بر روی ماشین اجرا می‌شوند دارد.

اکنون نظر عمومی بر این است که استفاده کنندگان باید بدون توجه به مجموعه دستورالعملهای ماشین، برنامه‌های خود را در زبانهای سطح بالا بنویسند. این نظریه با وجود جذابیت بسیار، در عمل درست از آب در نمی‌آید. برای مثال چنانچه به آگهیهای استخدام مراجعه کنید، خواهید دید که افراد آشنا به زبان اسمبلی ماشینهای مختلف خواستار آن زیادی دارند.

تحقیقات و نشریات فنی نیز همین تصویر را منعکس می‌سازند. به هنگام ارائه یک کامپیوتر جدید، طراحان غالباً درباره ملاحظات که به انتخاب آن طرح

منجر گشته است به بحث می‌نشینند، اما به ندرت اتفاق می‌افتد که در مورد دلایل طرح و پیاده کردن مجموعه دستورالعملهای انتخاب شده توضیح دهند. یکی از دلایل این امر احتمالاً این است که خود طراحان نیز تصور روشنی در باره اینکه چه دستورالعمل‌هایی را می‌خواهند در کامپیوترشان داشته باشند ندارند. در نتیجه، مجموعه دستورالعملها معمولاً ترکیبی است از تمام مجموعه‌های قبلی که بر روی مدلهای قدیمی تر آن شرکت تولید کننده قرار داشته و دستورالعمل‌هایی که قبلاً حذف شده ولی مجدداً به ضرورت آنها پی برده شده است.

مجموعه دستورالعملها یکی از عوامل مؤثر در موفقیت کامپیوتر است و باعث می‌شود تا سیستمهای عامل، برنامه‌های مترجم و سایر سیستمهای نرم‌افزاری، و نیز زبانهای برنامه‌سازی بتوانند با کارآئی و سهولت بیشتری پیاده شوند. بنابراین، چنین احساسی می‌شود که باید ملاحظات دقیقی در مورد طراحی و انتخاب مجموعه دستورالعملهای کامپیوتر و نیز تصمیماتی که در پیاده‌کردن آنها مؤثرند وجود داشته باشند.

۲- معیارهای طراحی مجموعه دستورالعملها

در طراحی و پیاده کردن مجموعه دستورالعملهای کامپیوتر باید عوامل زیر را در نظر گرفت: ۱- سازگاری، ۲- گداه و وضعیت، ۳- اندازه، مجموعه دستورالعملها، و کارآئی. اینک به بررسی هریک از این عوامل می‌پردازیم.

۲/۱ - سازگاری

در طراحی مجموعه دستورالعملها، سازگاری یکی از عوامل مطلوب است و در پیاده کردن آن، قطعاً نقش اساسی دارد. در طراحی باید ذوق و سلیقه به کار برد و آن را بر اساس طبیعت کار انجام داد. اگر زیرمجموعه‌ای از دستورالعملها نمایانگر سطح تجزیه نسبتاً بالایی باشد، کل مجموعه دستورالعملها نیز خود باید چنین باشد.

نا سازگاری در پیاده کردن می‌تواند به بروز مسائلی جدی بیانجامد. برای مثال، در مورد گداهای وضعیت بر روی ماشینهای پی دی پی ۱۱، نا سازگاری (خصوصاً به هنگام اجرای عملیات هم ارز) آثار ضعیفی نامطلوبی دارد، مشکلاتی را در اشکال زدایی برنامه به وجود می‌آورد، و بر پیچیدگی مترجم‌نویسی می‌افزاید.

۲/۲ - گداهای وضعیت

در اینجا باید به مطلوبیت معماری گداهای وضعیت صریح اشاره کرد. با وجودی که رایل معتقد است که به کار بردن بیشه‌ای وضعیت صریح، معماری متقارن تر و روشنتری

را نتیجه می‌دهد، اما اثر آنها را بر روی کارکرد کلی کامپیوتر مورد تردید می‌داند. آمار او استفاده بسیار کمی را از گداهای وضعیت نشان می‌دهد. به علاوه، او اشاره می‌کند که با توجه به میزان بالاسری لازم در پیاده کردن سیستم برای مقدار دادن به این بیتها، کارآئی سیستم تنزل می‌کند. اگرچه در سه سیستم بسیار متداول (یعنی آی بی ام ۳۶۰ و اخلاف آن، مونورولا ۶۸۰۰ و ۶۸۰۰۰، و پی دی پی ۱۱) از گداهای وضعیت صریح استفاده شده است، ولیسی نویسنده به دو دلیل با نتیجه‌گیریهای رایل موافق نیست:

(آ) اختلاف بین آمار ایستا و پویا در مورد حکمهای "مقایسه" و "جهش".
(ب) نقشی که بیتهای وضعیت در پیاده کردن عملیات محاسباتی ایفا می‌کنند. آمار رایل ایستا هستند: تحقیقات نشان می‌دهند که فراوانی پویای دنباله‌های از دستورالعملها که پس از محاسبه یک عبارت انضغاب می‌کنند (مثل احکام WHILE/DO و IF/THEN) به نسبت بیشتر است. از لحاظ پیاده کردن، دنباله‌هایی از ریزدستورالعملها که برای پیاده کردن عملیات محاسباتی نوشته می‌شوند، غالباً گداهای وضعیت را بررسی می‌کنند. در نتیجه در نظر گرفتن بیشه‌ای گد وضعیت در معماری داخلی سیستم ضرورت دارد. چنانچه این گداهای وضعیت در معماری خارجی نیز منعکس شوند (یعنی با آنهایی که در اختیار استفاده کننده قرار دارند یکسان باشند)، هیچ نوع بالاسری وجود نخواهد داشت و در نتیجه، وجود این بیتها باعث تنزل کارآئی نخواهد گردید.

۲/۳ - اندازه و محتوای مجموعه دستورالعملها

در حال حاضر، معماران کامپیوتر در ارتباط با اندازه و محتوای مجموعه دستورالعملها یکی از دو کار زیر را انجام می‌دهند:

(آ) به کارگیری مجموعه دستورالعملهای کوچک و کامل: چنین مجموعه‌ای از لحاظ علمی با ارزشتر است و پیاده کردن آن نیز کم هزینه‌تر می‌باشد. معماری ماشینهای پی دی پی، سری سایبر، چنین است.

(ب) انتخاب مجموعه دستورالعملهای شامل تمام دستورالعملهای ممکن: در بیشتر ماشینهای جدید، چنین گرایشی وجود دارد. پیاده کردن ماشینهای که دارای مجموعه دستورالعملهای کوچک، اما مناسب و کامل باشند کم هزینه‌تر و کار با آنها آسانتر است. هر چند تعریف "مناسب" و "کامل" اهمیت اساسی دارد و در غیر این صورت، ماشین در یک پا چند زمینه دچار نقصان در توانایی یا کارآئی می‌گردد. برای مثال، سیستمهای سایبر پی دی سی معماری کلاسیک ورق بزنید

آن به زمان بیشتری نیاز دارد. این امر غالباً "اتفاق می افتد". در غیر این صورت می توان روش اصولی تر و منسجم تری را در پیش گرفت.

۳ - یک روش سیستماتیک

آنچه که در زیر پیشنهاد می گردد، روش طراحی مجموعه دستورالعملها برای یک کامپیوتر همزمان با طراحی سخت افزار مربوطه می باشد. این روش شامل مراحل زیر است:

۱ - تحلیل کاربردهای عمده ای که در نظر گرفته شده اند.

۲ - شناسایی خصوصیات محیط کاربردها و برنامه هایی که در آن اجرا خواهند شد و مشخص ساختن عناصر ابتدائی و اصلی محیط کاربردها.

۳ - تعریف مجموعه ای از ریزدستورالعملهای عمودی در سطح بسیار پائین.

۴ - پیاده کردن عناصر ابتدائی و اصلی محیط کاربردها.

۵ - تعیین "وزن" هریک از عناصر ابتدائی سطح پائین و تعریف سخت افزار مربوطه.

۶ - مطالعه زیررشته های مشترک حاصل از مرحله ۴ و ایجاد مجموعه دستورالعملها. **مرحله ۱:** کاربردهای عمده ای که در نظر گرفته شده اند تحلیل می شوند. برای مثال آیا کامپیوتر اصولاً "برنامه های فزاینده و کوبول را اجرا می کند؟" آیا ماشین است برای اجرای زبانهای با ساختار بلوکی، یا ماشین است برای کار با پایگاه داده ها، و یا کامپیوتری است برای کنترل فرآیندها؟ این گونه تحلیل ها محیط کاربردها را مشخص می سازند.

مرحله ۲: پس از تعیین کاربردهای عمده ای که در نظر گرفته شده اند، باید خصوصیات محیط کاربردها و انواع برنامه هایی که در آن اجرا خواهند شد را برای دومین مورد شناسایی قرارداد: تعیین مجموعه ای از عناصر ابتدائی و ذاتی در محیط کاربردها که به نام عناصر ابتدائی سطح بالا یا کار بردی محیط خوانده می شوند، و تعیین میزان تکرار وقوع این عناصر ابتدائی در محیط کاربردها. در جدول ۱، نتایج برخی مطالعات آماری درباره زبانهای کوبول، فزاینده و پی ال وان به طور خلاصه نشان داده شده اند. اگر مجموعه عناصر ابتدائی در محیط کاربردها را با $\{h_1, h_2, \dots, h_m\}$ نشان دهیم، در این صورت، آمار فوق الذکر را می توان به شکل $\{wh_1, wh_2, \dots, wh_m\}$ تشریح نمود. مقدار wh_i نشانگر این است که هرچندگاه یکبار، عنصر ابتدائی h_i در مقایسه با سایر عناصر در برنامه ها اجرا می گردد. از روی چنین آمار می توان مجموعه دستورالعملهای مطلوب را برای رده

این امر شامل ترجمه برنامه های استفاده کننده به کدهای اجرایی کارآمد و نیز اجرای سریع دستورالعملهای ماشین می باشد. این سرعت و کارآیی به هردو عامل طراحی مجموعه دستورالعملهای ماشین و پیاده کردن کارآمد آن بستگی دارد. در نتیجه، اهمیت در اختیار بودن مجموعه دستورالعملهای مناسب مورد تأکید مجدد قرار می گیرد. همچنین باید توجه داشت که دستورالعمل کارآمدی برای یک سخت افزار بخصوص ممکن است برای سخت افزاری دیگر کارآیی نداشته باشد.

دستورالعمل نمونه فوق (دوران بیته به سمت راست) با فرض در اختیار بودن سخت افزار که بتواند دوران به سمت راست را برای هر تعداد از مواضع بیتها در یک دور انجام دهد، در نظر گرفته شد. چنانچه سخت افزار برای دوران به سمت راست هر موقع بیت به طور جداگانه یک چرخه پردازنده مرکزی را صرف نماید، مفید بودن این دستورالعمل مورد تردید قرار می گیرد. البته میانگین مقدار دوران بر حسب بیت در هر بار استفاده از چنین دستورالعملی کمتر یا مساوی ۷ باشد، در این صورت هنوز کارآیی آن دوبرابر بیش از ترکیب و است. از طرف دیگر، چنانچه طول کلمه کامپیوتر بزرگ باشد (۴۸ بیت یا بیشتر)، در این صورت و به وضوح از لحاظ اقتصادی با صرفه تر از تمام گروه دستورالعملهای دوران و تغییرمکان می باشند.

آنچه که گفته شد مهمترین مسأله را آشکار می سازد: جدا کردن طراحی و پیاده کردن یک مجموعه دستورالعملهای بهینه برای رده خاصی از کاربردها از توانائیهای سخت افزار مربوطه. اگر غیرممکن نباشد، بسیار مشکل است. اما امروزه، گروهی از مهندسان به طراحی سخت افزار (شامل مولفه های پردازش محاسباتی و منطقی، عناصر کنترل، مسیریابی جریان داده ها در داخل واحد سخت افزار، و مجموعه ریزدستورالعملها) می پردازند، گروهی دیگر از مهندسان مجموعه دستورالعملهای ماشین را طراحی می کنند، و گروه سوم مجموعه اخیر را با استفاده از ریزدستورالعملها پیاده می نمایند. این سه فرآیند طراحی و پیاده کردن تقریباً "به گونه ای مستقل از یکدیگر صورت می گیرند".

در نتیجه، کاری که انجام می شود تقلید^{۱۵} ماشین مورد نظر بر روی سخت افزار است که طرح شده است و نه طراحی سخت افزار به صورتی که برای مجموعه دستورالعملهای مورد نظر مناسب باشد. مثلاً سفارشی کردن روش هنگامی که سخت افزار مربوطه از قبل موجود باشد، اجتنابناپذیر است. به دلیل آنکه ایجاد سخت افزار به نسبت طراحی مجموعه دستورالعملها و پیاده کردن

و متناسب و توانائیهای محاسباتی بالایی داشتند، اما تا هنگامی که دستورالعملهای دیگری به مجموعه دستورالعملهایشان اضافه نشدند، از لحاظ کار با نویسه ها^۸ فوق العاده ضعیف بودند.

گاه افزودن دستوری به مجموعه دستورالعملها موجب افزایش کارآیی ماشین می گردد. به عنوان مثال، فرض کنید ماشینی با مجموعه دستورالعملهای حداقل، حاوی دو دستورالعمل عمومی برای کار بارشده بیتها و نویسه ها باشد. اولین دستورالعمل عبارت است از $EXTRACT OP_1, OP_2, OP_3, OP_4$ که بیتها را از OP_1 به صورتی که توسط OP_2 و OP_3 مشخص شده، استخراج می کند و رشته حاصله را که از سمت راست هم تراز^۹ شده و از سمت چپ نیز با مقدار فر پر شده است در OP_4 قرار می دهد. دومین دستورالعمل عبارت است از

$INSERT OP_1, OP_2, OP_3, OP_4, OP_5$

که رشته بیتهای از سمت راست هم تراز شده را از OP_1 به صورتی که توسط OP_2 مشخص شده است می گیرد و در مواضعی از OP_5 که توسط OP_3 و OP_4 مشخص شده قرار می دهد.

فرض کنید که ماشین ما ۱۶ بیتی است و کد ۱۶ بیتی دستورالعمل و نیز نشانی ۱۶ بیتی هر عملوند در یک کلمه حافظه قرار دارد. در این صورت برای اجرای عمل "دوران n بیتی به سمت راست"، با استفاده از یک حکم $EXTRACT$ و یک حکم $INSERT$ ، به ۲۴ چرخه ماشین نیاز می باشد. دقت کنید که حکم $EXTRACT$ به ۸ بار خواندن حافظه برای کد دستورالعمل، نشانی، و داده ها و نیز به یک دستبایی برای ذخیره کردن نتیجه و یک چرخه حافظه برای انجام خود عمل نیازمند است. حکم $INSERT$ نیز به ۱۰ بار خواندن حافظه، یک دستبایی برای ذخیره کردن نتیجه، و یک چرخه حافظه برای اجرای خود حکم نیاز دارد.

حال فرض کنید که ماشین برای دوران n بیت به سمت راست دارای حکم

$ROR OP_1, OP_2, OP_3$

به شرح زیر باشد. محتوی OP_1 به صورتی که توسط OP_2 مشخص شده دوران می یابد و نتیجه در OP_3 قرار می گیرد. این عمل تنها نیازمند ۸ چرخه حافظه (یعنی ۶ بار خواندن، یک دوران، و یک ذخیره کردن) است. واضح است که در مورد اول، عمومیت و زیبایی دستورالعملها، هزینه سه برابر را بر ما تحمیل کرده است. سوالی که باقی می ماند این است که چگونه چنین مجموعه دستورالعملهای کوچک و در عین حال مناسبی را طراحی و پیاده کنیم.

۲/۴ - کارآیی

کارآیی و سرعت اجرای برنامه های استفاده کنندگان حائز کمال اهمیت است.

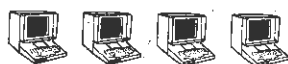
جدول ۱ - تکرر وقوع احکام اجرائی در زبانهای سطح بالای متداول.

پی‌اِل و ان	فهرست مرتب‌شده				کتابخانه				احکام اجرائی	
	(تحقیق دوم)		(تحقیق اول)		(تحقیق سوم)		(تحقیق دوم)			
ایستا	پویا	ایستا	پویا	ایستا	پویا	ایستا	پویا	ایستا	پویا	
ADD					۱۹/۸	۷/۷	۱۰	۶	۱۳/۷	۵/۴
IF					۳۳/۱	۱۸/۲	۳۰	۱۸	۲۴/۵	۱۵/۵
GO TO					۱۰/۶	۱۶/۵	۱۱	۱۹	۲۱/۷	۲۰/۳
MOVE					۲۶/۹	۳۳/۶	۳۰	۳۳	۱۲/۴	۲۷/۷
انتساب										
CALL										۰/۲
RETURN										۰
PERFORM					۶/۲	۱۳	۷	۸	۴/۱	۵/۴
DO										
CONTINUE										
READ										۲
WRITE										۲/۳
سایر احکام					۳/۴	۱۱	۵	۱۱	۸/۹	۱۱/۶

جلوگیری می‌کند. این قبیل تغییرات نوعی عدم تعادل را در معماری و سازماندهی سیستم به وجود می‌آورند. در هر حال، آنچه مسلم به نظر می‌رسد این است که مجموعه دستورالعملها را نباید بر سخت‌افزار موجود تحميل کرد بلکه باید آن‌ها را با هم و در یک زمان طراحی نمود. کامپیوتر با مجموعه دستورالعملهای تقلیل یافته (موسوم به ریسک ۱)، که در دانشگاه برکلی طراحی شده است، همین فلسفه را دنبال می‌کند. *

یادداشتها و مراجع

1. Marovac, N., "How to Design a Computer Instruction Set," Computer Bulletin, pp. 4-8, June 1983.
- 2 - سازگاری Consistency
- 3 - گدهای وضعیت Condition codes
- 4 - تجرید Abstraction
5. Russell, R.D., "The PDP-11: A Case Study of How Not to Design Condition Codes," Proc. of the Fifth Annual Symp. on Computer Architecture, ACM- SIGARCH, Apr. 1978.
- 6 - بالاسری Overhead
- 7 - ریزدستورالعمل ... Microinstruction
- 8 - نویسه Character
- 9 - هم تراز Justified
- 10 - تقلید Emulation
- 11 - بسامد Frequency
12. RISC = Reduced Instruction-Set Computer
- 13 - انتساب Assignment



$$Wp_j = \sum_{i=1}^m wh_i * wh_j p_j$$

با محاسبه هر Wp_j ، می‌توان اهمیت نسبی هر ریزعنصر ابتدائی را تعیین نمود. این عامل می‌تواند به عنوان راهنمایی در تعیین نیازهای پیاده‌کردن هر عنصر ابتدائی مورد استفاده قرارگیرد. به علاوه، عوامل Wp_j محلی را که باید در طراحی و پیاده‌کردن سخت‌افزار مربوطه مورد توجه قرار داد و نیز مجموعه ریزدستورالعملهای (افقی) واقعی مورد لزوم جهت پیاده‌کردن مجموعه دستورالعملهای ماشین را نشان خواهند داد. **مرحله ۶:** تحلیل عناصر زیررشته‌های ابتدائی (عمودی) سطح پائین، زیررشته‌های مشترک مرتبط با دستورالعملها که بیش از همه به کار می‌روند را مشخص می‌یازد. با مطالعه هریک از این زیررشته‌ها و تبدیلیهای حاصل از آنها بر روی داده‌ها، می‌توان طبیعت دستورالعمل واقعی مربوط به هر زیررشته را تعیین کرد. این روش مجموعه دستورالعملهای بهینه $\{u_1, u_2, \dots, u_n\}$ را نتیجه می‌دهد. حال می‌توان این مجموعه دستورالعملها را با استفاده از سخت‌افزار بهینه به دست آمده در مرحله ۵ پیاده نمود.

۴ - نتیجه‌گیری

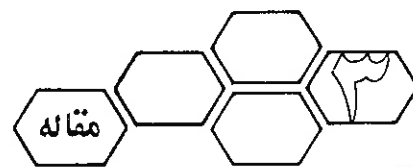
روش ارائه شده در فوق نه ساده است و نه کم هزینه، اما روش طراحی از بالا به پائین و سیستماتیک را با شیوه پیساده کردن از پائین به بالا برای یک مجموعه دستورالعملها ارائه می‌کند. هزینه اضافی این روش در آغاز مرحله طراحی سخت‌افزار قابل توجه است، چون از اعمال تغییرات پس از ساخت و بازاریابی کامپیوترها

خاصی از کاربردها تعیین کرد. البته مجموعه دستورالعملهای حاصل از این مرحله، بهترین مجموعه ممکن نخواهد بود و چهار مرحله دیگر در پیش هستند.

مرحله ۳: در اینجا می‌توان مجموعه‌ای از ریزدستورالعملهای عمودی را در سطح بسیار پائینی تعریف نمود. سطح این مجموعه باید خیلی پائین باشد و ریزدستورالعملهای موجود در آن باید مستقل از یکدیگر تا عمودی باشند. این مجموعه از ریزعناصر ابتدائی را با $\{p_1, p_2, \dots, p_n\}$ نمایش می‌دهیم.

مرحله ۴: حال باید عناصر ابتدائی محیط کاربردی یعنی $\{h_1, h_2, \dots, h_m\}$ را با استفاده از ریزعناصر ابتدائی یعنی $\{p_1, p_2, \dots, p_n\}$ پیاده کرد. این فرآیند، رشته‌هایی از ریزعناصر ابتدائی را که نمایانگر عناصر ابتدائی محیط کاربردی هستند تولید می‌کند. در این مرحله، می‌توان آمار ارتباط دهنده عناصر ابتدائی محیط کاربردی و ریزعناصر ابتدائی را تهیه نمود. به عبارت دیگر می‌توان بسامدی مانند $fh_j p_j$ را تعیین کرد. این بسامد نشانگر تعداد دفعاتی است که ریزعنصر ابتدائی p_j در پیاده‌کردن عنصر ابتدائی محیطی h_j به کار گرفته شده است. از اینجا به بعد، کار در دو قسمت مستقل از هم ادامه می‌یابد. **مرحله ۵:** تعیین کننده سخت‌افزار بهینه مربوطه، شامل مجموعه ریزدستورالعملهای واقعی (افقی) برای پیاده کردن مجموعه دستورالعملهای بهینه و مرحله ۶ تعیین کننده مجموعه دستورالعملهای بهینه است.

مرحله ۵: وزن هر ریزعنصر ابتدائی سطح پائین به ترتیب زیر تعیین می‌گردد:



نوعی مفسر زبان بیسیک با دقت محاسباتی زیاد

ترجمه آزاد
بهروز پرهامی و میترا کاوتزبان

با کاربرد روزافزون ریزکامپیوترها، زبان بیسیک به تدریج به صورت متداولترین زبان برنامه‌سازی کامپیوتر درآمده است. مترجمهای این زبان با نامهای گوناگون عرضه شده‌اند که از "بیسیک ۸۰"، "کام ۱"، "سی بیسیک"، و "سی بی ۸۰" می‌توان به عنوان کاملترین و پر قدرتترین آنها نام برد. در این چهار نوع زبان بیسیک، دقت عملیات بر روی اعداد نمایش داده شده بسیار متمایز است. تا ۱۴ رقم است. حدود توان برای دو زبان اول بین ۳۸-۱۰ و ۳۷-۱۰ و در دو زبان دیگر بین ۶۴-۱۰ و ۶۳-۱۰ می‌باشد. البته این حدود دقت برای بسیاری از کاربردهای تجاری و علمی کافی است، اما اخیراً نوع دیگری از بیسیک نیز موسوم به "بیسیک دقیق" یا "پی بیسیک" برای استفاده در عملیاتی که به دقتهای بالاتری نیاز دارند، عرضه شده است. دقت این نوع مفسر بیسیک تا ۱۲۲ رقم اعشار و حدود توان آن تا ۱۰۲۵۰ است.

مفسر "پی بیسیک" از نظر مشخصات عمومی فرق چندانی با مفسرهای دیگر زبان بیسیک ندارد. اما این مفسر در مقایسه با مفسرهای قویتر دارای محدودیتها و نقاط ضعفی است. مثلاً نامهای متغیرهای مورد قبول این مفسر باید تک حرفی و یا متشکل از یک حرف و یک رقم باشند. همچنین، متغیرها نمی‌توانند مقادیر رشته‌ای (حرفی) را اختیار کنند و این امر بخصوص در کاربردهای تجاری تولید اشکال می‌کند.

چون در "پی بیسیک" فقط یک نوع متغیر وجود دارد، تعریف متغیرها به صورت کسری یا صحیح بی مورد است. این مفسر اعداد را به صورت BCD (دو رقم در هر بایت) ذخیره می‌کند و با اینکه دقت قراردادی تا ۸ رقم اعشار است، اما می‌توان با دادن بازامتری در فرمان PRECISION، میزان دقت را تغییر داد. پارامتر فرمان PRECISION در واقع مشخص‌کننده تعداد بایتهای لازم برای ذخیره کردن عدد و بزرگترین مقدار قابل قبول برای آن ۶۱ است.

در هر نقطه برنامه می‌توان با دستوری مانند $P2 = PRECISION$ ، دقت کنونی محاسبات را ذخیره نمود، آن را با فرمان $P3 = PRECISION$ تغییر داد، و سرانجام در

هنگام نیاز مجدداً با استفاده از دستور $P2 = PRECISION$ دقت محاسبات را به مقدار اولیه بازگرداند.

برای نگهداری میزان دقت و با محدودیت تغییرات هر عدد، بایست جداگانه‌ای همراه با هر عدد ذخیره می‌شود. محتوی این بایست برای اعداد صحیح (کاملاً دقیق) مساوی صفر و برای اعداد کسری نشان دهنده دانه‌ها تغییرات (دقت) آن است. با انجام محاسبات مختلف، دقت نتایج به طور خودکار تعیین شده و همراه آنها ذخیره می‌شود.

دقت کنونی یک عدد با استفاده از تابع RANGE به دست می‌آید و توابع MAX و MIN برای تعیین حد فوقانی و تحتانی تغییرات عدد قابل استفاده‌اند. تابع MID یک مقدار نادقیق را به یک مقدار دقیق تبدیل می‌کند و عملگر ترکیبی $+/ -$ دقت یک عدد را به عدد دیگر منتقل می‌کند.

به طور کلی در برنامه‌سازی، مقایسه مقادیر کسری می‌تواند مشکلاتی را به وجود آورد؛ دو عددی که قاعدتاً باید مساوی در بیابند، ممکن است به خاطر خطاهای جزئی محاسباتی، با هم فرق داشته باشند. در زبان "پی بیسیک" این موضوع به صورت خودکار در نظر گرفته می‌شود. مثلاً در صورتی که محدوده تغییرات دو عدد $401/004$ و $401/001$ مساوی $0/005$ باشد، دو عدد با یکدیگر مساوی در نظر گرفته می‌شوند، چون عدد اول در واقع مقداری بین $400/996$ و $401/006$ و عدد دوم مقداری بین $400/997$ و $401/007$ را نشان می‌دهد و این دو محدوده دارای قسمت مشترکی هستند.

در "پی بیسیک" می‌توان عبارتی چون $1/3$ (یک سوم) را نیز مانند اعداد به عنوان اطلاعات ورودی به کامپیوتر داد. مقدار عدد "پی" با دقت ۱۲۲ رقم در داخل سیستم ذخیره شده و در هنگام نیاز قابل استفاده است. قالب خروجی با تابع ITEM مشخص می‌شود که سه پارامتر دارد: پارامتر اول مقداری است که باید چاپ شود، پارامتر دوم تعیین‌کننده نوع عدد است (صفر برای عدد صحیح، ۱ برای عدد با ممیز شناور، و ۲ برای عدد با ممیز شناور و توان)، و پارامتر سوم تعداد ارقام بعد از ممیز را نشان می‌دهد. این تابع معمولاً همراه با فرمان PRINT به کار می‌رود.

مفسر "پی بیسیک" یک محمول نرم‌افزاری کاملاً غیرعادی است. این مفسر برای کار بردهای داده‌پردازی تجاری چندان مناسب نیست، اما اگر به دقتی بیش از آنچه که در مفسرهای معمولی بیسیک وجود دارد نیاز داشته باشید، این مفسر را که تحت سیستم عامل سی بی ام ۴ کار می‌کند، مفید خواهید یافت.

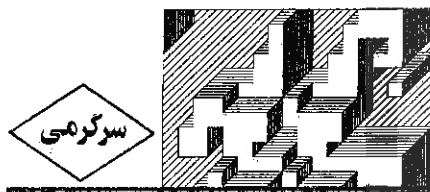
یادداشتها

- ۱- کسکام Bascom
- ۲- بیسیک دقیق PBasic = Precision Basic
- ۳- عملگر Operator
- ۴- سی بی ام CP/M ..

از "اینترپرایس رایج"، ژوئن ۱۹۸۳



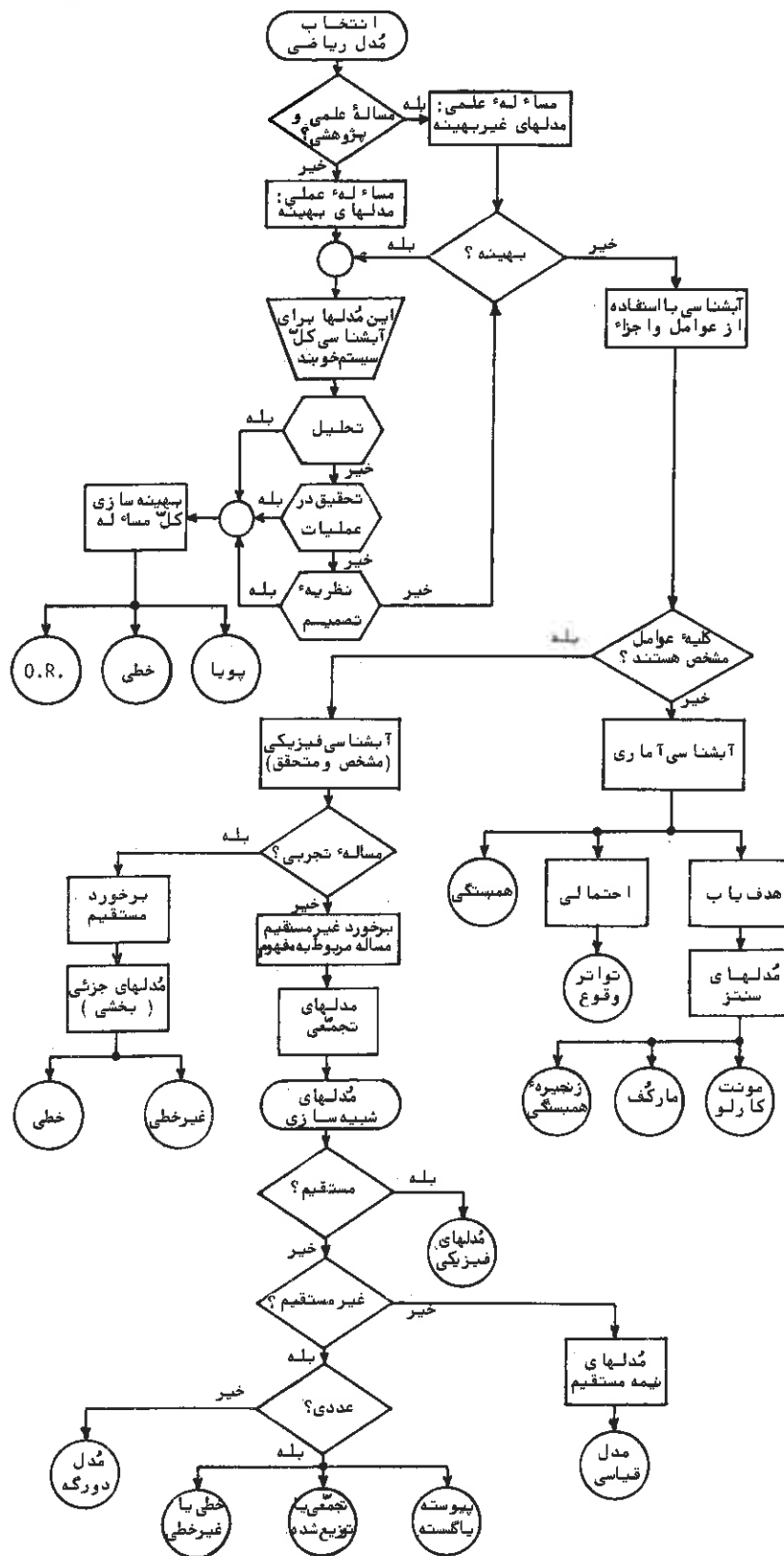
به سادگی می‌توان نشان داد که هر سال دی مورگان از حاصل ضرب دو عدد صحیح متوالی به دست می‌آید (مثلاً $42 \times 43 = 1806$ ، $43 \times 44 = 1892$ ، $44 \times 45 = 1980$). بنابراین سال دی مورگان بعدی سال ۲۰۷۰ میلادی خواهد بود ($2070 = 45 \times 46$). اگر فرض کنیم که هیچ کس پیش از ۱۳۰ سال عمر نمی‌کنند، آخرین سال دی مورگان سال ۱۶۷۰ در قمرن یکم دوشمست و هشتم میلادی خواهد بود ($1670 = 41 \times 42$). اما تا آن موقع، کسی چه می‌داند؟! *



پاسخ سرگرمی "سالهای دی مورگان"

گفتیم که سالهای ۱۸۰۶، ۱۸۹۲، و ۱۹۸۰ میلادی سالهای دی مورگان هستند. به این معنی که افراد متولد شده در این سالها می‌توانند (مانند ریاضیدان انگلیسی آگوستوس دی مورگان که در سال ۱۸۰۶ به دنیا آمد) در یکی از سالهای بعدی زندگی خود، چیزی درباره آن سال در رابطه با سن خودشان بگویند. دی مورگان در سال ۱۸۴۹ میلادی ۴۳ سال داشت. تساوی $43 = 1849$ جواب ممتاً را مشخص می‌کند. کسی که در سال ۱۸۹۲ به دنیا آمده باشد، در سال ۱۹۳۶ میلادی ۴۴ ساله بوده است ($44 = 1936$). به همین ترتیب، کسانی که در سال ۱۹۸۰ به دنیا آمده‌اند، در سال ۲۰۲۵ میلادی ۴۵ ساله خواهند شد.

ورق بزنید



نمودار گردش ۲ - روشهای مختلف مدلسازی برای منابع آب.

در سالهای اخیر، تعداد زیادی از مدل‌های ریاضی جعبه سیاه در زمینه‌های مختلف به وجود آمده و بسط یافته‌اند. تا آنجا که به تحولات داخلی جعبه‌های سیاه مربوط می‌شود، هنوز قوانین اصولی که به چنین تحولاتی منجر می‌شوند به صورت نظری یا تجربی کشف نشده‌اند. به بیان دیگر، استفاده از مدل‌های مجتمع، مکانیزم پایه‌ای تحولات آشناسی را تشریح نکرده و در نتیجه کمک شایانی به شناخت پدیده‌ها نمی‌کند.

در مدلسازی به صورت سیستم توزیع شده، تحولات داخلی مدل مورد تجزیه و تحلیل قرار می‌گیرند. در این حالت، سیستم در یک نقطه فضای مجتمع نیست، بلکه در نقاط مختلف و پراکنده (با در سطوح مختلف) بخش شده است. مختصات فضایی دارای اهمیت بوده و مدل ریاضی در واقع تمامی اجزای داخل سیستم را شبیه سازی می‌نماید. چنین مدلی اطلاعات مربوط به هر نقطه مورد نظر از سیستم را به دست می‌دهد.

اگر مدل ساز بخواهد رابطه بین بارش و جریان را با مدل توزیع شده شبیه سازی نماید، باید به قوانین و اصول اولیه هیدرودینامیک رجوع کند. به بیان دیگر، هرگاه لازم باشد که به قلب مسئله رجوع شود، باید از مدل‌های توزیعی سیستم‌های آشناسی استفاده نمود. تعداد متغیرهای مستقل مدل‌های توزیع شده بیش از یک بوده و این علاوه بر متغیر همبستگی، یعنی زمان، است. درحالی که مدل‌های جعبه سیاه دارای یک متغیر مستقل هستند.

بنابراین به عنوان مثال، مدل توزیع شده را می‌توان مجموعه‌ای از معادلات دیفرانسیل با مشتقات جزئی دانست، درحالی که مدل جعبه سیاه مجموعه‌ای از معادلات دیفرانسیل معمولی است.

هرگاه معادلات ریاضی که معرف مدل‌های آشناسی هستند از نوع غیرخطی باشند، سیستم آشناسی و مدل آن غیرخطی و در غیر این صورت، خطی است. در یک مدل خطی، مثل واحد هیدروگراف، خروجی مستقیماً "با ورودی متناسب است، حال آنکه این رابطه لزوماً برای مدل‌های غیرخطی صادق نیست.

۳ - سیستم‌های احتمالی و مدل‌های هدف یاب

بنا به تعریف، سیستم‌های آشناسی و مدل‌های آنها "معین" هستند اگر در هر زمان برای یک ورودی واحد و معین، عکس العمل یگانه سیستم همان باشد که از قبل معین شده است. در صورتی که عکس العمل سیستم در مقابل ورودی مشخص نامتعلق باشد، سیستم آشناسی و مدل آن را "احتمالی" می‌گویند. در یک سیستم آشناسی احتمالی، عوامل مؤثر نامتعلق ممکن است به دلایل ناشناخته‌ای تغییر نمایند. در این صورت فرض می‌شود که تغییرات عوامل مؤثر نامتعلق از قوانین

حساب احتمالات پیروی می کنند. بنا به تعریف اگر بین وقوع رویدادهای عوامل مؤثر نامتحقق یک رابطه زنجیره ای احتمالاتی وجود داشته باشد، سیستم آشناسی را "استوکاستیک" می نامند. در این مورد می توان از نظم احتمالی در رویداد عوامل مؤثر نامتحقق برای ساخت مدل هدف یسبب استفاده نمود.

تحقیق برای ساخت مدلهای هدف یاب در منابع آب هنوز در مراحل ابتدائی است. این مدلهای در واقع پیشرفته ترین بخش مدل سازی را در آشناسی نشان می دهند و هم اکنون در مراکز پژوهشی متعددی در سراسر جهان تحت بررسی قرار دارند.

۴- کاربرد کامپیوتر در مدل سازی

مسائلی که امروزه در مدلهای آشناسی مطرحند عملاً بدون استفاده از کامپیوتر قابل حل نیستند. بخش مهمی از این مسائل به تحقیق در عملیات مربوط می شود. حل این قبیل مسائل و نیز مطالعه مدلهای آشناسی در مقیاس بزرگ با استفاده از روشهای دستی نیاز به چندین میلیون نفر در ساعت کار دارد. اما با استفاده از کامپیوترهای نوین و روشهای تحقیق در عملیات، مانند برنامه ریزی خطی، این مسائل ظرف چند دقیقه حل می شوند. کاربردهای اصلی کامپیوتر در آشناسی به قرار زیرند:

(ت) پردازش داده ها به صورت متعارف.
(ب) حل معادلات مدلهای ریاضی تشریح کننده سیستمهای آشناسی.
(پ) کنترل ابزارهای علمی و آزمایشهای آشناسی.
(ت) حل مدلهای ریاضی بفرنج که با روشهای دستی غیر قابل حلند.

در مطالعات منابع آب، هم از کامپیوترهای قیاسی و هم از کامپیوترهای رقمی استفاده می شود. مدلهای قیاسی منابع آب معمولاً از یک سری اجزای الکتریکی مانند دیودها، خازنها، و تقویت کننده های مرکب تشکیل شده اند و کامپیوتر با کمک این اجزای الکتریکی، سیستم آشناسی را شبیه سازی می نماید. مهمترین مورد استفاده کامپیوترهای قیاسی در منابع آب در بخش آبهای زیرزمینی است.

به طور کلی به واسطه نمایش سریع واکنش سیستم آبهای زیرزمینی در مقابل داده های معین، کارشناسان نظری و تجربی این رشته از منابع آب به استفاده از کامپیوترهای قیاسی راغب شده اند. اما سه نقطه ضعف این مدلهای، یعنی کم بودن سرعت، پائین بودن دقت، و بالابودن هزینه، باعث شده اند که علیرغم پیشرفتهای چشمگیر، این روشها دیگر چندان طرف توجه مدل سازان منابع آب نباشند.

برای برطرف نمودن این نقایص، در سالهای دهه ۱۹۷۰ میلادی از تلفیق کامپیوترهای قیاسی و رقمی، ترکیب جدیدی که به آن "سیستم کامپیوتری دورگه" می گویند برای مدلسازی به وجود آمد. کاربرد عمده مدلهای دورگه در مدلسازی آبهای زیرزمینی، مدیریت است پروژه ها، و آبخیزداری بوده است.

کامپیوترهای رقمی وظیفه اصلی حل مسائل آشناسی را برعهده دارند. این مسائل را باید در قدم نخستین توسط نظریه جریان شبکه ها یا هر نظریه مناسب دیگری مورد تحلیل دقیق قرار داد و سپس اجزای آن را با کمک روشهای متعدد آنالیز عددی، به مسائل عددی تبدیل نمود.

تحلیل شبکه برای انتخاب مدل مناسب بر چهار رکن اساسی تکیه دارد:

(ت) کوتاه ترین مسیر از مبدأ به مقصد.
(ب) حداقل طول اتصالات.
(پ) جریان حداکثر از منبع به مقصد.
(ت) برنامه ریزی و کنترل (پرت و سی پی ام).
در استفاده از روشهای مختلفی که آنالیز عددی برای حل مسائل مدلهای ارائه می کنند، مدل سازان معمولاً تمایل به استفاده از روشی دارند که در کتابخانه کامپیوتر مورد استفاده شان موجود باشد، زیرا نوشتن نرم افزارهای کلی برای حل مسائل مدلهای ریاضی به زمان و کوشش بسیار نیازمند است و در صورتی که این عمل قبلاً انجام شده باشد، دوباره کاری ضرورتی ندارد.

مدلهای کامپیوتری منابع آب روز به روز با استفاده از شیوه های نوین تکامل می یابند. به جز مدلهای هدف یاب، مدلهای نظریه صف نیز اخیراً در خدمت آشناسان قرار گرفته اند. هر روزه مدلهای عددی بسیاری در سراسر جهان ابداع می شوند که ساخت و بهره برداری از این مدلهای به غنای بیشتر صنعت آب خواهد انجامید.

یکی از معروفترین این مدلهای به وسیله مرکز مهندسی آشناسی در آمریکا ساخته شده است. این مدل به بخشهای متعددی تقسیم می گردد که هر بخش عهده دار یک قسمت از مطالعات می باشد. مسائلی چون مسدود رودخانه ای، رسوب، رابطه بارش و جریان، و جز آن، از موارد عمده ای هستند که این مدل در نظر می گیرد. این مدل هنوز هم کامل نشده است و دائماً بخشهای جدیدی به آن افزوده می شوند و یا بخشهای قدیمی آن مورد تجدیدنظر قرار می گیرند. خوشبختانه این نرم افزار در ایران وجود دارد و قسمتهای آن در دست بهره برداری است.

۵- بحث و نتیجه گیری

یکی از سوالاتی که همواره در مورد مدل سازی مطرح بوده، و پایه برآورد اولیه

مخارج مدل را تشکیل می دهد، مسأله زمان مدلسازی است. این زمان طبیعتاً به حجم مسأله، عمق مطالعات، و نیاز استفاده کننده به دخالت دادن عوامل متعدد بستگی دارد. هنوز هیچ روش تحلیلی جامعی برای یافتن زمان مدلسازی به دست نیامده و اطلاعات در این زمینه فقط مبتنی بر تجربیات گذشته مدلهای کمابیش مشابه می باشد.

برای ساخت یک مدل شبیه سازی سه رودخانه ای، شامل گره ها و ارتباطات دلخواه (از قبیل سدها و کانالهای رابط)، یک تیم مدلساز به چیزی در حدود چهار سال وقت نیاز دارد تا مدل نسبتاً کامل و بی نقصی که شامل بررسیهای اقتصادی نیز باشد ارائه نماید. در اینجا این سوال مطرح می شود که چگونه می توان هزینه ساخت چنین مدلهایی را توجیه نمود.

پاسخگوئی به این سوال چندان مشکل نیست: اولاً در بسیاری از موارد حل مسائل آشناسی بدون وجود این مدلهای مقدور نیست و ثانیاً این مدلهای فقط یک بار ساخته شده و سپس همواره مورد استفاده قرار می گیرند. این مدلهای به سهولت قابل حمل و نقل بوده و می توانند با پذیرش شرایط داده های جدید، اطلاعات لازم را در اختیار آشناسان قرار دهند.

امروزه این مدلهای بسیاری از کشورها به رایگان در اختیار دانشگاهها و موسسات آموزشی و پژوهشی قرار می گیرند.

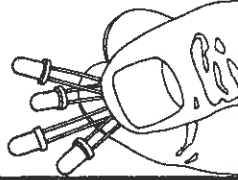
در مورد ایران می توان پیشنهاد نمود که یکی از مراکز مسئول مطالعات آب، به جمع آوری کلیه این اطلاعات پرداخته و کتابخانه ای از نرم افزارهای موجود ترتیب دهد. این کتابخانه سپس می تواند به عنوان مخزن اطلاعات در اختیار استفاده کنندگان مجازی که به آن نیاز دارند قرار گیرد. بدین ترتیب، می توان در مدت کوتاهی مطالعه همه جانبه ای را راجع به اوضاع آب ایران انجام داده و روشهای عملی و مناسبی برای حل مسائل آب یافت. *

پاداشتها

- ۱- مصالح کار Working medium
- ۲- جعبه سیاه Black box
- ۳- سیستم مجتمع Lumped system
- ۴- سیستم توزیع شده Distributed system
- ۵- معین Deterministic
- ۶- پرت و سی پی ام PERT & CPM
- ۷- نظریه صف Queuing theory

(تاریخ دریافت: ۲۵ اردیبهشت ۶۲)





نواوری‌ها

ضبط عمودی برای گرده‌های مغناطیسی

شرکت ژاپنی سونی ادعا می‌کند که به پیشرفتهای چشمگیری در زمینه ضبط عمودی اطلاعات بر روی گرده‌های مغناطیسی (میدان مغناطیسی عمود بر سطح گرده، به جای میدان مغناطیسی سطحی) دست یافته است. با این روش می‌توان تا ۴۵ میلیون بایت اطلاعات را بر روی گرده‌های لِرزان ۵/۲۵ اینچی و تا ۸ میلیون بایت را بر روی گرده‌های ۳/۵ اینچی ضبط نمود. گرده‌های مغناطیسی ۵/۲۵ اینچی که اکنون در بازار موجودند و از این روش ضبط استفاده می‌کنند گنجایشی در حدود ۱۰ میلیون بایت دارند. ★

از "ویکلی مایکرو بولتن"، ۱۶ آوریل ۱۹۸۳

یادآوری:

تتمید به موقع عضویت شما در انجمن موجب استمرار فعالیتها و خدمات انجمن خواهد شد.

تقاضا نامه تجدید عضویت، تغییر نشانی، اشتراک ماهنامه، و تجدید اشتراک

نام خانوادگی _____ نام _____

☐ تجدید عضویت برای یک سال.

نوع عضویت: _____ شماره عضویت: _____

☐ تغییر نشانی یا تلفن.

☐ اشتراک ماهنامه برای مدت یک سال.

☐ تجدید اشتراک ماهنامه برای یک سال.

شماره اشتراک: _____

نشانی: _____

شهر: _____ کدپستی: _____

شماره تلفن: _____

حق عضویت سالانه اعضا: عادی ۱۵۰۰ ریال (شامل اشتراک ماهنامه) وابسته ۱۵۰۰ دانشجوی ۵۵۰
حق اشتراک یک ساله گزارش کامپیوتر ۸۰۰ ریال
لطفاً "رسید پرداخت وجه به حساب جاری شماره ۱۱۱۷ انجمن نزد بانک صادرات شعبه ۱۶۵۲ (دانشگاه صنعتی شریف)، تهران، را به درخواست خود ضمیمه کنید. پرداخت وجه از طریق هریک از شعب بانک صادرات در تهران و شهرستانها ممکن است. برای دریافت بروشورهای فعالیتها و انجمن و شرایط عضویت با مکارانه تماس بگیرید.

آدمک مصنوعی برای خنثی سازی بمب

کاربرد آدمکهای مصنوعی توسط نیروهای پلیس، به منظور جلوگیری از خطرات جانی در هنگام خنثی سازی بمب و عملیات مغازده آمیز دیگر، سابقه ای شش ساله دارد. آدمک مصنوعی جدیدی که توسط یک شرکت کانادایی برای این منظور ساخته شده است، حدود ۱۰۰ کیلوگرم وزن دارد و بازوی ۱/۵ متری آن را می‌توان به وسایل مختلفی مجهز نمود. چرخها و شاسی این آدمک مصنوعی طوری طرح شده اند که آن را قادر به صعود از پله و عبور از لبه جدول خیابان می‌سازند. این آدمک مصنوعی را می‌توان به دوربین تلویزیونی و میکروفن نیز مجهز نمود و قیمت آن در حدود ۴۰ هزار دلار است.

از "نیوزویک"، ۲۸ مارس ۱۹۸۳

توانا شیهای کامپیوتر شخصی آی بی ام ساخت ژاپن

کامپیوتر شخصی چند استفاده کننده ای مدل ۵۵۵۰ آی بی ام که در ژاپن ساخته می‌شود دارای توانا شیهای بسیار جالبی است. این ماشین می‌تواند به عنوان یک کلمه پرداز فوق العاده قوی، با واژه نامه ای بسیار بزرگ و قابلیت انجام بررسیهای دستوری و نحوی، عمل نماید.

ژاپنها دارای دو روش خط نویسی هستند: یکی استفاده از الفبای "کاناکانا" که هر حرف آن یک سیلاب را نشان می‌دهد و کلمات از ردیف کردن این سیلابها تشکیل می‌شوند و دیگری استفاده از سیستم "کانجی" که دارای علامت ویژه ای برای هر کلمه است و به چیزی در حدود ۸۰۰۰ علامت مختلف نیاز دارد.

کامپیوتر شخصی فوق الذکر می‌تواند متون را به طور خودکار از یک روش خط نویسی به روش دیگر تبدیل نماید. نمایشگر به کار رفته در این سیستم بسیار دقیق است و علائم را در ماتریشهای نقطه ای ۱۶x۱۶ یا ۲۴x۲۴ نشان می‌دهد.

قیمت حداقل این سیستم در حدود ۴۲۰۰ دلار است که شامل پردازنده ۸۰۸۶، حافظه ای به گنجایش ۲۵۶ کیلوبایت، نمایشگر غیر رنگی، یک گرده لِرزان ۲۵۶ کیلوبایتی، و چاپگری با سرعت ۶۰ علامت در ثانیه می‌شود. با دوبرابر کردن اندازه حافظه، جایگزین نمودن یک نمایشگر دقیق تر، و گرده سخت ۸/۱ میلیون بایتی، قیمت سیستم به ۱۲۸۰۰ دلار می‌رسد. نرم افزارهای موجود بر روی این سیستم شامل سیستم عامل "داس" و زبانهای فرترن، کوبول، و پاسکال است. ★

از "ویکلی مایکرو بولتن"، ۱۶ آوریل ۱۹۸۳

محصولات جدید

سرعتترین آتِرتکا میپوترجهان

شرکت فوجیتسو ژاپن در سال ۱۹۸۲، ساخت دو آتِرتکا میپوتر جدید خود را به نامهای "فاکوم وی پی ۱۰۰" و "فاکوم وی پی ۲۰۰" اعلام نمود. این دو آتِرتکا میپوتر با قدرت اجرای ۵۰۰ میلیون عمل ریاضی بر روی اعداد با ممیز شناور در ثانیه، بسیار سریعتر از مدل قبلی همین سری به نام "فاکوم ۲۳۰-۲۵۰ وی پی ۲" که قدرت اجرای ۲۲ میلیون عمل ریاضی در ثانیه را داشت، هستند. این دو آتِرتکا میپوتر در مقایسه با "کری ۱"، که تا چندی پیش به عنوان سرعتترین آتِرتکا میپوتر جهان شناخته می‌شد، نیز ۶ تا ۷ برابر پُر قدرت تر هستند.

تفاوت بین مدلهای وی پی ۱۰۰ و وی پی ۲۰۰ در گنجایش حافظه آنها است، به طوری که اولی ۱۲۸ و دومی ۲۵۶ میلیون بایت حافظه دارد. این ظرفیت عظیم حافظه در مقایسه با ظرفیت رایج آتِرتکا میپوترها که تا ۳۲ میلیون بایت است، افزایش چشمگیری را نشان می‌دهد. برای حافظه این آتِرتکا میپوترها برای نخستین بار از تراشه های ماس ایستا ۴ با ظرفیت ۶۴ کیلوبیت استفاده شده است. البته هنوز شرکت فوجیتسو به طور رسمی موفقیت خود را در زمینه ساخت این نوع حافظه ها اعلام نکرده است، ولی عرضه این دو آتِرتکا میپوتر می‌تواند به طور ضمنی نشان دهنده این موفقیت باشد.

سرعت چشمگیر این دو آتِرتکا میپوتر مرمون استفاده از حافظه های مرموز و نیز مدارهای مجتمع در مقیاس بزرگ با سرعت فوق العاده زیاد است. این مدارهای مجتمع که شامل ۱۳۰۰ تا ۱۴۰۰ درجهه منطقی هستند، دارای تاه خیر زمانی معادل ۳۵۰ پیکوثانیه می‌باشند.

شرکت فوجیتسو این ابرکا میپوترها را از سال ۱۹۸۳ میلادی به بازار عرضه خواهد کرد و پیش بینی می‌کند که طی پنج سال اول، تعداد ۳۰ دستگاه از آنها را به فروش برساند. برای این منظور، شرکت فوجیتسو تاه کید زیادی بر صادرات خواهد داشت.

1. FACOM VP-100
2. FACOM 230-75APU
3. Cray-1
4. MOS static RAM

از "فیس اگویپمنت اند پروداکشن"،

شماره ۴، سال ۱۹۸۲