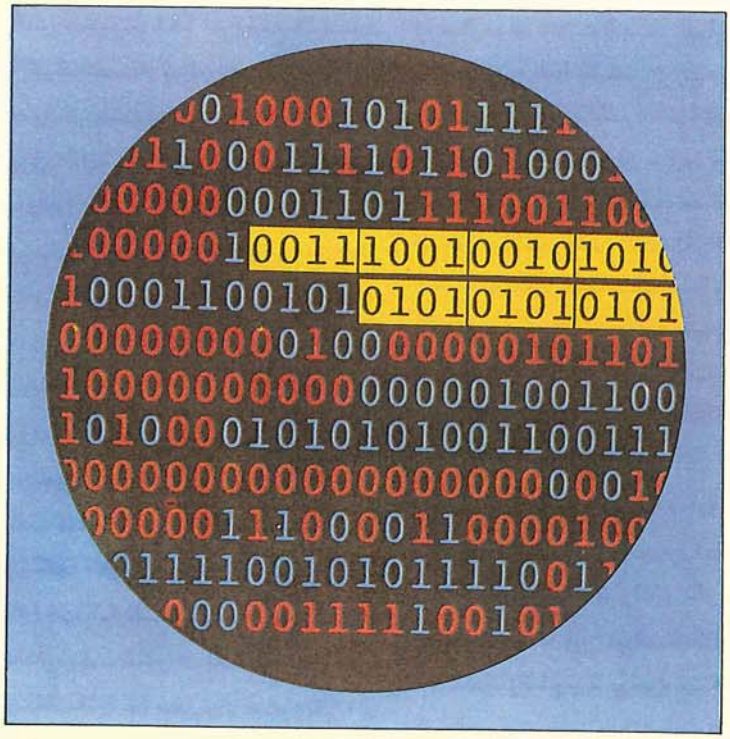
لغات البرمجة

د. أحمد شرف الدين أحمد

لغات البرمجة هي وسيلة الاتصال والتخاطب بين الإنسان والحاسب والتي عن طريقها يقوم الإنسان ببرمجة الحاسب الآلي لأداء أعمال معينة، فهي بذلك أساس ولا غنى عنها في مجال الحاسبات، فالحاسب كجهاز لايمكن الإستفادة منه ما لم تتم برمجته، ويسمى الشخص الذي يقوم ببرمجة الحاسب باسم مبرمج.



مرت لغات البرمجة خلال مراحل تطورها بأطوار متعددة وذلك بداية من مرحلة لغات الآلة ونهاية بلغات الجيل الرابع. ومما يجدر ذكره أن مرحلة الجيل الرابع لا تعني نهاية تطور لغات البرمجة، إذ قد ظهرت لغات أكثر تطورا مع حاجات العصر المتجددة والمتطورة دوما.

تنقسم لغات البرمجة إلى أربعة أقسام رئيسة هي:-

١ ـ لغة الآلة: وهي نظام ثنائي ذو أرقام لها دلالة تحتاج إلى شخص يتعلمها جيدا وتحتاج إلى أوامر كثيرة كما أن احتمالات الخطأ فيها كبيرة .

٢ ـ لغة التجميع: ويمكن فيها استبدال الأوامر المختلفة برموز دالة عليها كما أصبحت برامجها أسهل من برامج لغة الآلة، ويشار إليها ـ وكذلك إلى لغة الألة ـ بلغة البرمجة منخفضة المستوى Low Level Programming Languages وذلك لقرب هذه الأوامر من أوامر الآلة. واللغة تختلف من حاسب إلى آخر فترجمة لغة الآلة تحتاج إلى برنامج يسمى المجمع ، من أمثلتها لغة Plan ولغة Neat وقد تم تطوير لغة التجميع بادخال أوامر مركبة (Macro Assembly) متداخلة،إلا أن الاعتماد الكلي على الآلة وصعوبة تذكر الأوامرمن أبرز مساوىء هذه اللغة.

٣ ـ لغة الأداء العالي: وتعتمد على ترجمة المعادلات(Formula Translation)، من أمثلتها لغة الفورتران التي تعنى بالتطبيقات العلمية والهندسية، والتي تطورت إلى فورتران ٤ ثم إلى فورتران ٦٦ وبعدها ونتيجة لازدياد أهمية البرمجة الهيكلية ظهر فورتران ٧٧ الذي تطور إلى فورتران 8 X. يمر البرنامج في هذه اللغة عبر ثلاث مراحل قبل التنفيذ، وهي مرحلة الترجمة (Compilation) حيث تترجم الأوامر بلغة فورتران إلى لغة الآلة، ومرحلة الربط (Linking) حيث يتم دمج البرامج المعروفة ووضع ملف لها، وأخيرا مرحلة تحميل البرنامج في ذاكرة الحاسب للتنفيذ. ويمكن استخدام لغة الأداء العالي في كثير من التطبيقات.

٤ ـ لغة الجيل الرابع: الغرض منها تخفيف العبئ على المبرمج، وقد ظهرت لرفع إنتاج البرمجيات وزيادة نسبة الصيانة للبرامج الجديدة وللحاجة الماسة لمزيد من المبرمجين واستخدام قواعد البيانات (Data Bases) والنظم العاملة على التوازي (Parallel System)، وقد ساعدت على توفير وقت المبرمج، إضافة إلى أنها تعمل على أجهزة متعددة.

تطبيقات لغة الأداء العالي

أمكن الاستفادة من لغة الأداء العالي في برمجة كثير من التطبيقات العملية يمكن إبراز بعضها في التالي:-

١ ـ التطبيقات العلمية والهندسية

كانت الاستخدامات الأولى للحاسبات وقفا . لى الحاسبات العلمية ، ومن أجل ذلك تم إبتداع لغة الفورتران لتيسير مهمة البرمجة وجعلها لا تعتمد على آلة بعينها وكان التركيز الأساس في لغة الفورتران على كفاءة البرنامج التنفيذي المولد حيث أن معظم المبرمجين آنذاك كانوا يعتقدون أن المترجم لن يستطيع توليد برنامج كفء مثلما يستطيع المبرمج لو أنه كتب برنامجه بلغة الآلة أو التجميع مباشرة.

أمكن في عام ١٩٥٨م وضع اللبنات الأولى للغة تمتاز بالوضوح ووجود قواعد يمكن صياغتها، وتم تسميتها بلغة الألجول (Algol) وهي اختصار لكلمتي (Algorithmic Language)، وظهر أول تقرير رسمي شامل لهذه اللغة في يناير ١٩٦٠م يعرف بـ (Algol - 60). وأهم مايميز لغة الألجول هو تنظيم بنيان البرنامج وتحاشي بعض المساوىء التي أُخذت على لغة الفورتران مثل التعريف الآلي للمتغيرات وعدم وجود كلمات محفوظة في اللغة .

تعد لغة الألجول ـ رغم أنها لم تنتشر بكثافة كلغة الفورتران ـ الأساس لكل لغات البرمجة الهيكلية التي تلتها ومن أبرزها لغة الباسكال. وقد تطورت لغة الألجول مثل سائر لغات البرمجة فظهرت عام ١٩٦٨م الألجول ٦٨ (Algol - 68)، ورغم إنتشار هذه اللغة في التطبيقات العلمية والهندسية إلا أن لغة الفورتران مازالت مستخدمة في برمجة هذه التطبيقات.

٢ ـ لغات التطبيقات التجارية

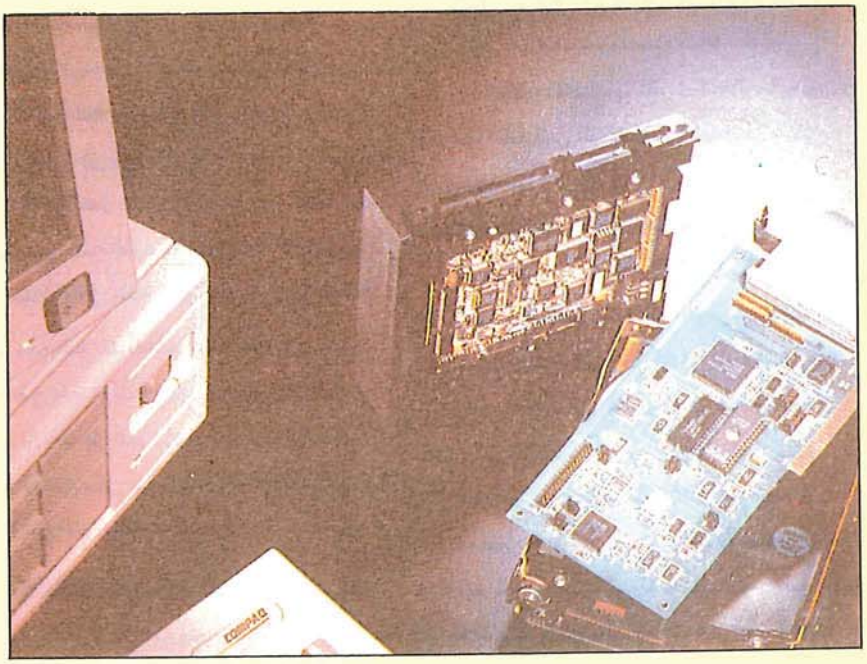
يقصد بالتطبيقات التجارية هنا تلك التطبيقات الخاصة بالمحاسبة وادارة الأعمال. وقد ظهرت الحاجة إلى لغة برمجة خاصة بهذه التطبيقات مع بدء استخدام الحاسب في هذه المجالات والتي يصعب برمجتها بلغة الفورتران ـ أولى لغات الأداء العالي ـ كما أن لها صعوبة في تشكيل المدخلات والمخرجات (Input / Output)، وفي الواقع فإن أولى اللغات الخاصة بالتطبيقات التجارية من لغات الأداء العالي هي لغة الكوبول Cobol والتي إشتق إسمها من Common Business Oriented Language.

بدأ وضع اللبنات الأولى للغة الكوبول عام١٩٥٩م حينما إرتأت وزارة الدفاع الأمريكية أن هناك ضرورة لابتكار لغة خاصة بالأغراض التجارية يمكن استخدامها مع الأنظمة الإلكترونية المختلفة، وقام بوضع مواصفات هذه اللغة مؤتمر لغات أنظمة البيانات (The Conference On Data System Languages)

والتي يرمز لها إختصارا بالإسم CODASYL.

ظهرت أولى مترجمات لغة الكوبول بعد ذلك بعدة سنوات ومن ثم وضعت أولى مواصفاتها القياسية عام ١٩٦٨م وهو ما يسمى بكوبول ٦٨. وفي عام ١٩٧٤م ظهرت لغة الكوبول ٧٤.وأخيراً في عام ١٩٨٥م ظهرت كوبول ٨٥ . تتميز لغة الكوبول باستخدام مفردات اللغة الإنجليزية في أوامرها مما يسهل على المبرمج تذكرها ويجعل البرنامج سهل القراءة والتتبع، كما أن هذه اللغة بها أوامر للفرز وكتابة التقارير بصورة مفصلة وهي إمكانات لا تتوفر في غيرها من لغات البرمجة المشابهة، من الجانب الآخر فإن من مساوىء لغة الكوبول كثرة الكلمات وتعدد الأوامر التي تقوم بنفس الوظيفة كما أنها تحدد مناطق معينة على السطور لكتابة الأوامر المختلفة.

ومما يجدر ذكره أنه بالرغم من أن الكوبول هي أولى لغات البرمجة للأغراض التجارية فإنها مازالت الأكثر استخداما وانتشارا مقارنة باللغات المنافسة والتي طورت لذات الغرض من لغات الأداء العالي. ونظراً لتعدد التقارير التي تتطلبها النظم التجارية إرتأت شركة (IBM) في منتصف الستينيات عمل لغة خاصة باستخراج التقارير وهي ما تعرف بلغة آر بي جي (R P G) والمشتقة من Report Program Generator. وربما كان الدافع لذلك آنذاك هو أن استخراج التقارير بلغة الكوبول يتطلب برنامجا طويلا كما أن معظم التقارير لها نفس المواصفات العامة. وقد لاقت هذه اللغة في بادىء الأمر بعض النجاح وتم تطويرها إلى ما يعرف بلغة (RPG 11) وكذلك (RPG 111) والتي أضافت بعض الإمكانات الرياضية إلى اللغة الأصلية. وقد قامت بعض الشركات بكتابة مترجمات لهذه اللغة على آلاتها مثل شركة DEC , HP ولكن من الملاحظ أن استخدامات وإنتشار هذه اللغة حالياً ضئيل جداً مقارنة بلغة الكوبول، كما أنه بإضافة إمكانات التقارير إلى بنية لغة الكوبول فإن الميزة التي كانت تتمتع بها لغة آر بي جي قد تلاشت.



● الحاسب الآلي وتطبيقاته العلمية .

٣ ـ التطبيقات العلمية والتجارية

ويقصد بها لغات البرمجة التي تم تطويرها لتفي بمتطلبات التطبيقات العلمية والتجارية في آن واحد. وقد بدأ التفكير في تطوير لغة واحدة تجمع مزايا لغتي الفورتران والكوبول في إطار واحد منذ وقت مبكر. بدأت شركة (IBM) هذه المحاولات في عام ١٩٦٤م تحت إسم New Programming Language ، وبعد ذلك في عام ١٩٦٥م تم تغيير إسم هذه اللغة إلى (PL/1) وهي إختصار لـ (Programming Language 1) ، ومن مزايا هذه اللغةسهولة الترميز وسعة المرونة ، أما مساوئها والتي كانت سببا في عدم إنتشارها ومن ثم إندثارها فهي صعوبة التعلم بالنسبة للمبرمجين المبتدئين.

وعلى العكس من لغة (PL/1) فإن لغة البيسك (BASIC) قد لاقت نجاحا عظيما كلغة سهلة الإستعمال للأغراض العلمية والتجارية، ويأتي اسم تلك اللغة من إختصار كلمة Beginners All Purpose Symbolic Instruction Code.

ويدين الإنتشار السريع لهذه اللغة إلى إنتشار الحاسبات المصغرة والحاسبات الشخصية والتي إتخذت من هذه اللغة أساسا لبرمجتها خاصة في الأيام الأولى لإنتشارها. وتمتاز لغة بيسك بسهولة التعلم والترميز كما أنها تعطي إمكانات واسعة في التطبيقات العلمية تضاهي تلك التي تقدمها لغة الفورتران، إضافة إلى ذلك فإن لها إمكانات جيدة في التطبيقات التجارية يمكن عن طريقها تنفيذ غالبية هذه التطبيقات بسهولة معقولة وإن كانت تلك الإمكانات لاترقى بحال إلى تلك التي تقدمها لغة الكوبول .

عادة ما تُقدَّم نظم تحويل لغة البيسك إلى لغة الحاسب في صورتين هما مفسِّر ومترجم يقوم بترجمة البرنامج الأصلي كله مرة واحدة ومن ثم تنفيذه.

٤ ـ برمجة النظم

إزدادت أهمية برمجة النظم (System Programming) بازدياد تعدد نظم التشغيل التي يتم تطويرها، وفي أولى مراحل التطوير تم برمجة نظم التشغيل (Operating Systems) باستخدام لغة التجميع للآلة التي يعمل عليها نظام التشغيل. ومع تعقد نظم التشغيل وازدياد تعقد نظم البرمجة بصفة عامة ظهرت الحاجة لاتباع أسلوب جديد في البرمجة يعرف بالبرمجة الهيكلية (Structured Programming). وقد كان لهذه المبادىء العامة في طرق البرمجة أثر بالغ على وضع مواصفات لغات البرمجة التي ظهرت بعد ذلك وأيضا على تعديل مواصفات اللغات القديمة التي كانت موجودة من قبل. ولعل في ظهور لغة الفورتران ٧٧ أبلغ دليل على ذلك حيث سمحت بصفة رسمية بتعريف البنى الأساس في البرمجة الهيكلية.

تم تطوير مبادىء البرمجة المستخدمة في تعليم المبتدئين وترجمة نظم التشغيل. ومن أهم اللغات التي أستخدمت مبادىء البرمجة الهيكلية لتعليم البرمجة للمبتدئين ونظم التشغيل لغة باسكال (Pascal) التي يأتي اسمها على اسم العالم الفرنسي الشهير. وقد تم تطوير هذه اللغة بصفة مبدئية عام ١٩٧٠م بوساطة **نيكولاي ويرث**، وقد كانت الفكرة الأساس وراء هذه اللغة هي البساطة والوضوح مما جعلها إمتدادا للغة الجول ٦٠ حيث أزالت كل نقاط الضعف الأساس التي بها. وقد تم إضافة إمكانات أكبر لها في مجال تراكيب البيانات كما أن إحتوائها على كل مقومات البرمجة الهيكلية أكسبتها شهرة واسعة بين المبرمجين وأساتذة العلم. ومن المثير أن نذكر أن أول مترجم للغة باسكال قد تم كتابة معظمه بنفس اللغة.

ونظرا لأن برمجة النظم تكون معقدة للغاية إذا ما تمت بلغة التجميع بدلا من أي لغة من لغات الأداء العالي، فربما فقدت كفاءتها في التنفيذ لذا فقد تم تطوير لغات لها نفس التراكيب الأساس الموجودة في لغات الأداء العالي وتسمح في نفس الوقت بالتعامل حتى على مستوى الوحدات الثنائية (bits) وإجراء العمليات التي لا تتوافر إلا في لغات التجميع. ومن أشهر هذه اللغات لغة (C) والتي تستخدم ليس فقط في برمجة النظم بل في كتابة الكثير من نظم التطبيقات الأساس مثل برمجة منسق كلمات أو نظام إدارة قواعد بيانات أو ما إلى ذلك. وقد تم تطوير هذه اللغة في معامل شركة بل ضمن نظام التشغيل العالمي يونكس (UNIX) والذي يعد نظام التشغيل القياسي الوحيد حتى الآن. وبذلك تعد لغة (C) من اللغات التي لا تعتمد على آلة بعينها مما يكسبها إنتشارا وذيوعا وذلك على العكس من لغة مثل اس بي ال (SPL) وهي لغة برمجة هيكلية تسمح بالتعامل على مستوى لغة الآلة أيضا ولكنها لغة محلية خاصة بأجهزة (HP 300) فقط.

تطبيقات لغة الجيل الرابع

تعمل نظم لغات الجيل الرابع إما على جهاز معين أو على أجهزة عدة كما أنها يمكن أيضا أن تتعامل مع قاعدة بيانات داخلية خاصة أو عدد من قواعد البيانات. ومن أشهر لغات الجيل الرابع تلك التي تعمل على معالجة الحركة المستخدمة أساسا في التطبيقات التجارية وتشمل ما يلي:-

١ ـ نظام مابر

يعد نظام مابر (Mapper) ـ إختصار لـ (Maintaining, Preparing and Processing Executive Reports)

ـ أحد الوسائل المتطورة لتطوير تطبيقات تتم بسرعة وكفاءة عاليتين ودون الحاجة لمبرمجي تطبيقات. ولهذا النظام قاعدة بياناته الخاصة، ويمكن عن طريق هذا النظام توليد كميات لانهاية لها من التقارير بأبسط جهد ممكن، بل ويمكن كذلك عمل تطبيقات معقدة بدون الحاجة إلى اللجوء إلى لغات الجيل الثالث للبرمجة، كما أنه سهل حيث يكفي يومين إثنين فقط لتدريب المستخدم النهائي على استخدامه.وباستخدام هذا النظام يمكن للمستفيد إنشاء ملفات والتعامل معها وإجراء مختلف العمليات الحسابية والمنطقية وعمل أنواع مختلفة من التقاريرعلى الخط وبانتقاء الوظيفة التي يريدها من قائمة الخيارات (Menu) والتي تظهر للمستفيد على شاشة الوحدة الطرفية. توجد بالنظام إمكانية طلب المساعدة (Help facility) في أي نقطة داخل النظام. ومن عيوب هذا النظام ـ وهو أيضا عيب في معظم لغات الجيل الرابع ـ أنه يعتمد على جهاز معين وهو يونيسس (Unisys). كما أنه لا يمكنه التعامل مباشرة مع الملفات الأخرى وقواعد البيانات الأخرى حتى إذا كانت موجودة على نفس الجهازإذ أن هذا النظام له قاعدة بياناته الخاصة به. وعادة ما يتم التغلب على هذه الصعوبة الأخيرة بإنشاء ملفات مستوية يمكن بها تبادل البيانات بين نظام مابر وأي نظام آخر.

٢ ـ نظام رابيد

نظام رابيد (Rapid) هو أحد نظم لغات الجيل الرابع التي يستخدمها محلل النظم وأخصائي الحاسبات لعمل التطبيقات المطلوبة دون أن يستخدمها المستخدم النهائي. ويمكن عن طريق هذا النظام عمل قاموس البيانات وتحديث قاعدة البيانات وذلك أثناء مرحلة التصميم، ويمكن أيضا تكوين مجموعة معقدة من الإستفسارات باستخدام هذا النظام، وعند الحاجة إلى الإستفادة من إستخدام هذا النظام بوساطة المستخدم النهائي يجب توفر الشروط التالية:ـ

(أ) سهولة الإستخدام عن طريق القوائم على الوحدة الطرفية مع وجود إمكانية المساعدة عن طريق الحاسب .

(ب) ضرورة وجود قائمة من القيم المشتركة (Default Values) تمثل معظم طلبات المستفيد العادي بحيث لا يضطر لادخال بيانات عديدة متكررة في نواحي كثيرة. فعلى سبيل المثال يكفي لكي يطلب المستفيد استخراج تقرير به بيانات معينة أن يحدد أسماء الحقول المرادة، أما عملية تنسيق التقرير(Formatting) وعناوينه وعدد السطور بالصفحة وما إلى ذلك فتكون لها قيم مختارة معقولة مع الإحتفاظ بالمرونة الكافية للمستفيد لتغييرها إذا ما أراد.

(ج) أن تكون اللغة غير إجرائية بمعنى أن المستفيد يخبر اللغة بما يريد لا بخطوات الحصول على ما يريد، فمثلا إذا أراد أن يعرف متوسط درجات طلاب في مادة ما يكفي أن يعطى الأمر (Average grades) وتكون مسؤولية هذه اللغة تحديد كيفية حساب المتوسط.

(د) إمكانية إستخدام قواعد البيانات الموجودة أوالملفات الأخرى بسهولة ويسر، بحيث يظهر نموذج قاعدة البيانات في صورة ذات علاقة ببعضها البعض حتى إذا كانت هي داخليا خلاف ذلك. فعلى سبيل المثال فإن اللغة (Query/3000) والمستخدمة على جهاز إتش بي ٣٠٠٠ (HP 3000) يمكنها التعامل مع قاعدة البيانات الشبكية بحيث تبدو للمستفيد النهائي وكأنها قاعدة بيانات ذات علاقة بسيطة.

وبطبيعة الحال فإن هذه اللغة إذا كانت لاستخدام محلل الأنظمة أو المبرمج أو أخصائي الحاسبات عامة فإنها يجب أن تعطي بعض الإمكانات الأخرى الهامة وذلك مثل:ـ

- السماح بالتعامل من خلال لغات البرمجة الأخرى (الجيل الثالث) وذلك لتنفيذ التطبيقات الأكثر تعقيداً وصعوبة .
- السماح بإنشاء قاعدة معلومات عن البيانات وصياناتها وتحديثها (Data Dictionary).
- وجود إمكانية إختبار (Debugging) في داخل اللغة.
- السماح بأداء بعض العمليات التي عادة ما تتم عن طريق نظام التشغيل مباشرة .

في الواقع تغطي كثير من لغات الجيل الرابع إحتياجات الطرفين: أخصائي الحاسبات أو المبرمج أو محلل الأنظمة والمستفيد النهائي، وكل ما هنالك أن بعض الإمكانات تصبح متاحة للطرف الأول ولايستطيع الطرف الآخر إستخدامها.

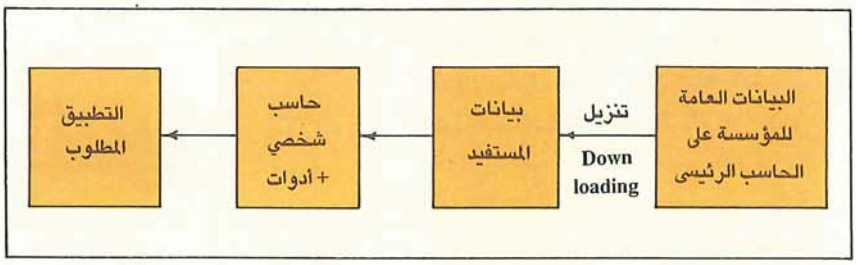
الدور الجديد للمبرمج

مضت تلك الأيام التي كان فيها المبرمج هو المسؤول عن برمجة كافة التطبيقات المطلوبة في مكان عمله والتي تميز بها عالم الحاسبات منذ نشأته حتى بداية الثمانينيات، ويمكن إيجاز طرق تطوير التطبيقات الحالية فيما يلي:ـ

١ ـ الحصول على حُزَم برامج أونظم جاهزة لأداء كل العمل المطلوب، وهو الأسلوب السائد الآن، كأن تشتري الجهة المعينة نظاماً جاهزاً للمحاسبة بوساطة الحاسب وكل المطلوب من المستخدم لهذا النظام هُو تزويده بالمتغيرات المختلفة التي تناسب متطلباته.

٢ ـ الحصول على أدوات تطوير سهلة

يمكن للمستخدم النهائي إستخدامها مباشرة أو بعد تدريب بسيط، وبهذا يمكن لهذا المستخدم أن يقوم بتطوير التطبيقات التي يريدها مباشرة . ويظهر هذا الآن في البرامج الشهيرة لتنسيق الكلمات أو لوحة الحاسبات أو قواعد البيانات البسيطة والتي تعمل على الحاسبات الشخصية، وبهذا يكون دور الحاسب الرئيس في الجهات الكبيرة التي تستخدم هذا النوع من الأجهزة هو توفير البيانات المطلوبة والتي بها يتمكن المستخدم النهائي من عمل تطبيقاته التي يحتاج إليها ، ويبين الشكل التالي مخطط لهذه العملية.



● مخطط إنتقال البيانات في الحاسبات .

٣ ـ التعاون بين محلل النظم والمستفيد النهائي لعمل التطبيق المطلوب مباشرة وبدون تدخل المبرمج لكتابة البرامج، وقد أصبح من الممكن إتباع هذا الأسلوب نظرا لتوفر أدوات تطوير النظم والتي يمكن لمحلل النظم أوخبير الحاسبات إستخدامها لعمل النظام المطلوب مع المستخدم النهائي. وهكذا فبدلا من أن يقوم محلل النظم بكتابة مواصفات البرامج كما هى العادة، فإنه يقوم بعمل التطبيق مباشرة، وتساعده على ذلك وسائل هندسة النظم (CASE) وهي إختصار (Computer Assisted Software Engineering)

وهذا لا يعني البتة إنتهاء دور المبرمج، ولكن هناك تغييرا ملحوظا في هذا الدور خاصة فيما يختص بدور مبرمجي التطبيقات. أما فيما يختص بالمبرمجين الذين يقومون بإنشاء الأنظمة الأساس فإن دورهم مازال قائما بل وتزداد الحاجة إليهم وخاصة ذوي المهارة العالية منهم حيث يقومون بتطوير أنظمة التشغيل والمترجمات والبرمجة العامة علاوة على إنشاء الأنظمة التجارية والتطبيقات الجاهزة لاستخدام المستفيد النهائي. ومن المهم التأكيد على أن أسلوب تطوير هذه الأشياء قد تغير تغيرا ملحوظا ، فعلى سبيل المثال فإنه من النادر أن يبدأ تصميم أمثال هذه النظم من فراغ بل عادة ما يتم الحصول على منتج يمكن البناء فوقه وإتمامه للحصول على المنتج النهائي. وهذا واضح في عمل مترجمات اللغات حيث تقوم مترجمات المترجمات (Compiler Compiler) بإنجاز معظم العمل المطلوب.

وثمة نوع آخر من المترجمين وهم مبرمجي النظم، وهؤلاء مازال لهم دور كبير في أنظمة الحاسبات الكبيرة وبعض الحاسبات الصغيرة ، وتنحصر المهمة الأساس لهؤلاء في التأكد من حسن إستغلال موارد الحاسب وتنغيم النظام للحصول على أقصى كفاءة ممكنة من الأجهزة المتوفرة. وفي الواقع فإن نظم التشغيل الحديثة تعطي مبرمج النظم إمكانية كبيرة وبيانات عديدة تفصيلية مما يجعل مهمته أيسر كثيرا من ذي قبل.

مستقبل لغات البرمجة

لقد تم إختراع الكثير من لغات البرمجة على مدى تاريخ إستخدام الحاسبات حتى الآن وهي تتفاوت تفاوتا بيِّنا من حيث إمكاناتها وتقبل المستفيدين منها لها. كما حدثت تطورات عديدة في مجال البرمجة في السنوات الأخيرة فظهرت لغات ذات نوعيات جديدة تتوائم مع التطورات التي حدثت في الأجهزة (Hardware) وكذلك النظم (Software)، فعلى سبيل المثال ظهرت لغات للتعامل مع أجهزة الحاسبات المتوازية وذلك للإستفادة من إمكانية تنفيذ الأوامر على التوازي مثل لغة (Occam) كما ظهرت لغات أخرى لإدارة شبكات الحاسبات وغيرها لتطبيقات الذكاء الإصطناعي مثل لغتي (Lisp) و(Prolog) وغيرها، وكذلك لأنظمة الوقت الحقيقي (Real -time) مثل لغة (Ada)، أو لأنظمة المحاكاة مثل لغة (GPSS)، أو للأنظمة الرياضية الكبيرة مثل لغة (Protran)، أو للتعامل مع أنظمة الرسم مثل لغة (HPGL)، أو للتعامل مع الذاكرة بحسب المحتوى وغيرها.

ومن المتوقع أن يزداد التطور في هذه اللغات الخاصة وأن تصبح مترجمات اللغات أكثر تسامحا مع المبرمج، وبمعنى آخر ستكون المترجمات أكثر ذكاءاً، كما أنه من المتوقع أن يستمر الإتجاه لإحداث لغات تخاطُب بين الإنسان والآلة بطريقة أقرب إلى اللغة الطبعية عما هي الآن.

وفيما يختص بالتعامل مع قواعد البيانات فإن لغات المستقبل سوف تحمل في طياتها إمكانات الإسترجاع الإستنتاجي (Deductive retrieval) بحيث يصبح بإمكان لغة البرمجة إستخراج بعض المعارف أو المعلومات غير الموجودة في قاعدة البيانات بصورة مباشرة. وكذا سوف يمكن التعبير في هذه اللغات ـ لمصممي ومبرمجي قاعدة البيانات ـ عن شروط صحة البيانات المجربة في قاعدة البيانات وتكاملها، وبذلك لا يكون من الممكن تخزين أي بيان خطأ في قاعدة البيانات يخل بتكاملها وذلك بطريقه آليه.