



البرامج الفرعية

(1) ما هو مفهوم البرامج الفرعية ؟ و ما مميزاتها ؟

البرامج الفرعية : هي أحد وسائل البرمجة لحل مشكلة كبيرة، أو هي تجزئة المشكلة الكبيرة إلى مشاكل أو مسائل أصغر، ثم برمجة هذه المسائل الصغيرة فرادى، ثم يكون البرنامج الأساسي الذي يقوم بربط أو تجميع وظائف هذه الأجزاء الصغيرة بالتتابع الذي يحل المسألة أو المشكلة الأساسية .

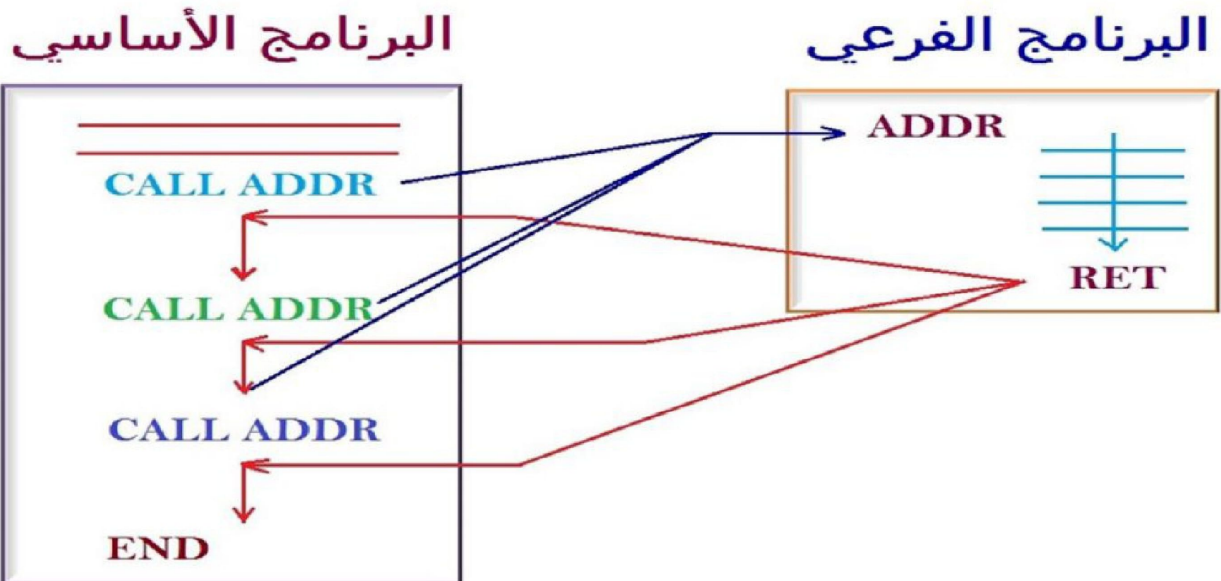
المميزات :

- 1 تسهيل عملية البرمجة .
- 2 اختصار كمية الذاكرة المستخدمة لكتابة شفرات البرنامج .

(2) كيف تبني البرامج الفرعية ؟

تبني البرامج الفرعية كغيرها من البرامج العادية و لكن عندما تنتهي تعود إلى النقطة التي بعد النقطة التي تفرعت عندها (تشبه إلى حد كبير طريقة تنفيذ برامج القفز) .

- 1 يتم النداء على البرنامج الفرعي باستخدام الأمر CALL فيقوم البرنامج الأساسي بتنفيذ البرنامج الفرعي .
- 2 عند الإنتهاء من التنفيذ يعود البرنامج بأمر RET إلى البرنامج الأساسي و إلى نفس المكان الذي خرج منه .



(3) ما الفائدة من استخدام البرامج الفرعية ؟

توفير الذاكرة المستخدمة لكتابة البرنامج : أي أنه يكون جزء من البرنامج نحتاج إلى تكرار كتابته أكثر من مرة في البرنامج الأساسي، فعند استخدام البرامج الفرعية يتم تكرار أمر النداء فقط بدلاً من تكرار البرنامج الفرعي ككل، مثال على ذلك برامج أزمنة التأخير التي يمكن كتابتها في هذه الحالة مرة واحدة بـ CALL .

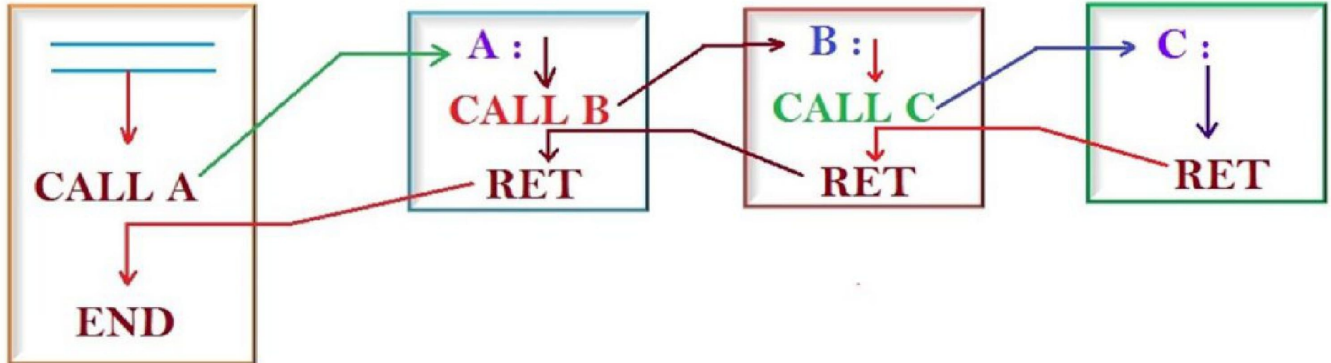
(4) ما هي المكدسة (Stack) ؟ و مؤشر المكدسة (Stack Pointer) ؟

المكدسة (Stack) : هي جزء مقتطع من الذاكرة RAM لخدمة أغراض النداء و العودة من البرامج الفرعية و أيضاً لخدمة أغراض المقاطعة و أغراض أخرى .

مؤشر المكدسة (SP) : هو مسجل مكون من 16 بت أو على حسب نوع المعالج، و محتوياته هي عنوان قمة أو آخر مكان تم التخزين فيه في المخزن، و عندما يكون المخزن فارغ فإن المؤشر يشير إلى قاعه، و عند الإضافة إلى المخزن فإن المؤشر ينقص محتوياته، و عند السحب فإنه تزيد محتوياته .

(5) ما هي عملية التداخل (Nesting) أثناء وجود أكثر من برنامج فرعي ؟

- عملية التداخل : أي أن أي برنامج فرعي يمكنه النداء على برنامج فرعي آخر .
- فمثلاً : البرنامج الأساسي ينادي على البرنامج الفرعي A، و البرنامج الفرعي A ينادي البرنامج الفرعي B، و البرنامج الفرعي B ينادي البرنامج الفرعي C، و هكذا لأي عدد التداخلات .



- بعد الإنتهاء من تنفيذ آخر برنامج فرعي في السلسلة (و ليكن البرنامج الفرعي C) :
- ✓ يرجع المعالج إلى البرنامج الفرعي B حيث تم النداء على البرنامج الفرعي C .
- ✓ تتم تكملة البرنامج الفرعي B حيث يرجع المعالج إلى البرنامج الفرعي A من حيث تم النداء على البرنامج الفرعي B .
- ✓ بعد الإنتهاء من تنفيذ البرنامج الفرعي A تتم العودة إلى البرنامج الأساسي من حيث تم النداء على البرنامج الفرعي A .

(6) ما هي تعليمات المناداة و العودة ؟

1 تعليمية المناداة (CALL) :

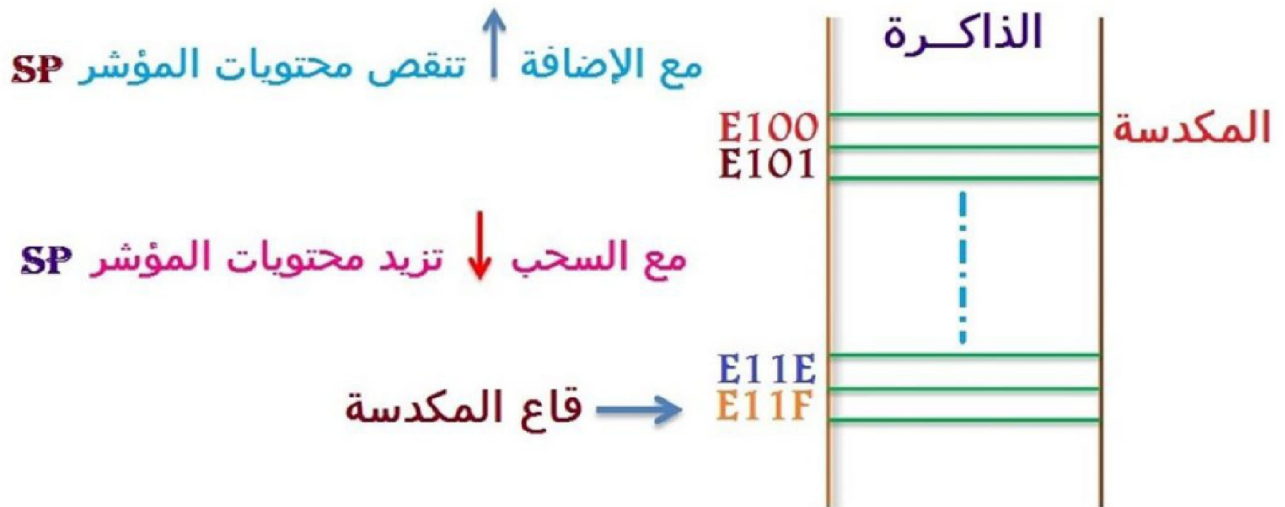
الأعلام المتأثرة	العملية	الصيغة	المعنى	الكلمة المختزلة
لا يوجد	يُتابع التنفيذ في البرنامج الفرعي من العنوان المحدد بواسطة المتحول Operand الموجود في تعليمية المناداة، و المعلومات المطلوبة من أجل العودة مثل : IP, CS تحفظ في المكس .	CALL Operand	مناداة برنامج فرعي	CALL

2 تعليمية العودة (RET) :

الأعلام المتأثرة	العملية	الصيغة	المعنى	الكلمة المختزلة
لا يوجد	—	RET/RET Operand	العودة إلى البرنامج المُستدعي	RET

(7) اشرح بالتفصيل كيف يعود البرنامج إلى المكان الذي خرج منه . مع توضيح دور المكس ؟

- السريكمين في المخزن (Stack) و مؤشر المخزن (Stack Pointer)، حيث :
- 1 عند النداء على البرنامج الفرعي فإن محتويات عداد البرنامج (PC) التي تحتوي على عنوان الأمر التالي في التنفيذ في البرنامج الأساسي يتم دفعها إلى حيث يشير مؤشر المكس (SP) و بالتالي تنقص محتويات المؤشر (SP) .
- 2 يتم تحميل عداد البرنامج بعنوان بداية البرنامج الفرعي و بذلك تنتقل عملية التنفيذ إلى هناك .
- 3 في نهاية البرنامج الفرعي يوجد الأمر RET الذي بتنفيذه يتم سحب قمة المخزن التي يشير إليها المؤشر و توضع في عداد البرنامج (PC) مرة أخرى .
- ✓ و بذلك تتم العودة إلى البرنامج الأساسي و عند نفس المكان الذي تم الخروج منه .



أسئلة الباب السادس

- ① ما هو مفهوم البرامج الفرعية ؟
- ② كيف تبنى البرامج الفرعية ؟
- ③ ما هي الفائدة من استخدام البرامج الفرعية ؟
- ④ ما هي عملية التداخل Nesting أثناء وجود أكثر من برنامج فرعي ؟
- ⑤ اشرح بالتفصيل كيف يعود البرنامج إلى المكان الذي خرج منه . مع توضيح دور المكدسة .