الباب الرابع

بيئة النوافذ

Windows Environment

مفتتح

نبدأ فى هذا الباب فى العبور من برامج الكونصول إلى البرامج النوافذية. والبرنامج النوافذى يتضمن الكثير من الأنشطة بخلاف الكود الذى يخدم التطبيق نفسه. والسبب فى ذلك أن البرنامج النوافذى ليس كائناً مستقلاً ، فهو يرث من بيئة النوافذ كل خصائصها ، كما أن الاتصال الدائم بينه وبين نظام التشغيل النوافذى يتضمن الكثير من التفصيلات. لهذا كله فإن الأستوديو يوفر عليك الكثير من هذه التفصيلات بمنحك خدمة الساحر الذى يكتب لك الكود الخاص بالإجراءات التنظيمية بحيث تستطيع أن تحول بؤرة تركيزك غلى المهمة التى يقوم بها التطبيق.

(٤-١) طرق برمجة التطبيقات النوافذية

هناك أسلوبين أساسيين فى إنشاء البرامج النوافذية:

- باستخدام دوال الوصلة البينية API (اختصار الاسم Application Programming Interface).
- باستخدام مكتبة مؤسسة الفصائل MFC (اختصار Microsoft Foundation Class).

والفارق الرئيسى بين الطريقتين أنك عندما تستخدم دوال الوصلة البينية فإن عليك أن تكتب كل سطور البرنامج بنفسك من الألف إلى الياء. ومع ذلك فإن الطرازات الحديثة من الأستوديو قد قدمت خدمة الساحر (Wizard) الذى يعفيك من كتابة الأجزاء المتكررة ويبنى لك الهيكل العام للبرنامج النوافذى. ولديك الاختيار أن تبدأ من الصفر أو تبدأ بهيكل مبدئى للبرنامج يبنيه لك الساحر. وسواء استخدمت الساحر أم لا ، فإن الأدوات التى تستخدمها فى البرنامج هى مجموعة من الدوال (بضعة مئات) بالمكتبة API.

أما مكتبة مؤسسة الفصائل فهى بمثابة واجهة ، تفصل بينك وبين دوال الوصلة البينية API. وهى تمنحك أساساً خاصية البرمجة الموجهة نحو الأهداف (OOP).

ومكتبة الفصائل كما يوحى اسمها عبارة عن شجرة من الفصائل تمثل الأهداف المختلفة فى بيئة النوافذ بدءاً من نافذة التطبيق الرئيسية إلى ما يتبعها من النوافذ الأبناء (Child Windows) مثل صناديق الحوار النوافذ القافزة إلى آخره ، وبداخل كل فصيلة

مجموعة من الدوال الأعضاء التى تتحكم بها فى سلوك الهدف. وإذا كنت قد استخدمت مكتبة الأهداف OWL (Object Windows) التى أنتجتها شركة بورلاند (Borland) من قبل ، فإن مكتبة مؤسسة الفصائل MFC تؤدى نفس الدور الذى تؤديه مكتبة الأهداف OWL لكنها تختلف عنها فى تصميم الفصائل وأسمائها. كما أن مكتبة مؤسسة الفصائل MFC لا تمنحك فقط مجموعة من الفصائل ، بل تمنحك خدمة جليلة وهى خدمة الساحر (MFC Wizard) الذى ينشئ لك الهيكل الرئيسى للبرنامج بكل ما يحتويه من إجراءات متكررة ، بحيث تصبح مهمتك هى إضافة بعض سطور الكود إلى البرنامج فى الأماكن المناسبة التى يقترحها عليك الساحر!

الطرازات القديمة من اللغة

إذا كنت قد استخدمت طرازا قديما من طرازات لغة سى المرئية (Visual C++) فمن الجدير بالذكر أن تلاحظ أن الطرازات القديمة من اللغة (من الطراز 1.0 حتى 2.5) كانت تعمل فى بيئة النوافذ التقليدية ونظام التشغيل دوس ، أى أنها تستخدم لإنتاج برامج ١٦ بت. أما الطرازات الجديدة من اللغة ، بما فيها الطراز 6.0 ، فهى تعمل فى بيئة نوافذ ٩٥ و ٩٨ و NT ، وهى جميعا ٣٢ بت. ولا تتوافق البرامج ١٦ بت مع البرامج ٣٢ بت لأن طريقة عنونة الذاكرة مختلفة تماما بكل منهما. ومن البديهى أن تصميم مكتبة مؤسسة الفصائل MFC يختلف فيما بين الطرازين.

ولهذا السبب فقد أطلق الاسم **Win32 API** على الوصلة البينية المستخدمة فى البيئة ٣٢ بت ، كما أطلق الاسم **Win16 API** على دوال الوصلة البينية المستخدمة فى البيئة ١٦ بت.

استدعاء دوال الوصلة البينية

ومن الجدير بالذكر أن دوال الوصلة البينية مكتوبة أصلا بلغة سى ، لكنك تستطيع استخدامها (استدعائها) من خلال أى لغة من لغات البرمجة مثل لغة بيسك المرئية أو لغة باسكال أو فورتران وبالطبع من لغة التجميع (Assembly).

برمجة العمليات المتعددة (Multitasking)

بقى أن تعرف أن بيئة النوافذ بأنواعها الثلاثة تستخدم خاصية المهام المتعددة (Multitasking) ، بمعنى أنها تنفذ أكثر من عملية (Process) واحدة فى نفس الوقت. ويتم ذلك بتقسيم وقت المعالجة على العمليات المختلفة بالتناوب ، ولأن الشريحة الزمنية المخصصة لكل عملية صغيرة جدا ، فإنه يبدو للإنسان أن العمليات تتم فى نفس الوقت. وقد كانت بيئة النوافذ التقليدية تتمتع ببعض خصائص العمليات المتعددة ولكنها كانت تتبع نظاما فقيرا فى إمكاناته وهو "Non-preemptive Multitasking" الذى يعتمد على تصميم البرامج نفسها وعلى قابليتها للمشركة فى وقت المعالجة. أما النظام الذى

قدمته نوافذ NT (والذى ينطبق على نوافذ ٩٥ و ٩٨ أيضاً) فيسمى "Preemptive Multitasking".

كما تتميز بيئة النوافذ أيضاً بخاصية الخيوط المتعددة (Multithreading) ، حيث يتم تقسيم العملية الواحدة إلى شرائح تسمى الخيوط (Threads) ؛ والخيط عبارة عن سلسلة من التعليمات تنفذ بطريقة متصلة واحدة تلو الأخرى. ومن المتبع أن يحتل البرنامج الواحد خيطاً واحداً ، ومع ذلك فإن بعض البرامج يمكن أن تحتل أكثر من خيط ، فيختص أحد الخيوط بالطباعة فى الخلفية مثلاً ، ويختص الأخر بالمهمة الأساسية للبرنامج. ولأن برمجة الخيوط تعتبر من المهارات الرفيعة (التى لا تخلو من تعقيد) فى مجال البرمجة ، فإن استخدام مؤسسة الفصائل تسهل عليك هذه المهمة.

(٤-٢) الأحداث فى بيئة النوافذ Events

هناك فارق جوهرى بين الطريقة التى يعمل بها البرنامج النوافذى (أى فى بيئة النوافذ) وبين برامج الكونصول أو البرامج المكتوبة لنظام التشغيل دوس. إن برنامج الكونصول هو المتحكم فى زمام الأمور وهو الذى يدير دفة العمل من لحظة التشغيل حتى نهاية التنفيذ. أما فى بيئة النوافذ فالأمر يختلف حيث تتولى بيئة النوافذ نفسها دفة القيادة ، أما دور البرنامج التطبيقى فينحصر أساساً فى استقبال **الرسائل (Messages)** الموجهة إليه من بيئة النوافذ والاستجابة لها.

وتعتمد بيئة النوافذ بصفة عامة على **الأحداث (Events)**. فأنت عندما تحرك مؤشر الفأر فوق نافذة ما أو تضغط على زر الفأر فى موقع ما من الشاشة ، فإن حدثاً يتولد ، وتشعر به بيئة النوافذ على الفور ، فتتعرف على نوع الحدث و موقعه ، كما تتخذ الإجراء المناسب للاستجابة لهذا الحدث. وعلى سبيل المثال فإنك لو حركت الفأر فوق شاشة أحد البرامج التطبيقية ، مثل وورد ، فإنك تلاحظ أن مؤشر الفأر يتغير شكله بحسب المنطقة التى يقع فيها. فأثناء عبوره فوق صفحة النص يأخذ شكل مؤشر الكتابة المميز بالخط الرأسى ، أما إذا مر فوق القوائم وسطور الأدوات فيأخذ شكل السهم ، فإذا انتقلت إلى سطر المهام (Task Bar) أو إلى حدود نافذة التطبيق الحالى فإنه يأخذ شكلاً جديداً وهكذا. أى أن هناك خط اتصال دائم ما بين كل التطبيقات المفتوحة ونظام التشغيل (النوافذ).

وتحصل النوافذ على المعلومات عن الأحداث من ثلاثة مصادر:

١. من أجهزة الدخل (Input Devices) مثل الفأر أو لوحة الأزرار.

٢. من الأهداف المختلفة فى بيئة النوافذ مثل القوائم (Menus) ، وأزرار سطور الأدوات (Toolbar buttons) ، وقضبان الانزلاق (Scroll bars) وأدوات التحكم الموجودة بصناديق الحوار (Dialog Boxes). ومن المفهوم أنك تولد الأحداث باستخدام الفأر أو لوحة الأزرار مع هذه العناصر ، ولكنه من وجهة نظر النوافذ ، فإن هذه العناصر هى مصدر الأحداث.

٣. قد يتولد الحدث من بيئة النوافذ نفسها وبدون تدخل منك ، فعلى سبيل المثال عندما تغطى إحدى النوافذ نافذة أخرى وتخفى محتوياتها ، فإن بيئة النوافذ تتعرف على هذا الحدث (وهذا الحدث عبارة عن انتهاء صلاحية (Invalidate) محتويات النافذة الواقعة فى الخلفية).

وتستجيب النوافذ للأحداث بإرسال الرسالة المناسبة إلى التطبيق الذى جرى به الحدث. ففى المثال الأخير فإن النوافذ ترسل رسالة إلى البرنامج التطبيقى الواقع فى الخلفية لتخبره بأن محتويات النافذة قد انتهت صلاحيتها ، ويلزم إعادة طلائها بالمحتويات من جديد. ولو أنك ضغطت بالفأر على النافذة التى انتهت صلاحية محتوياتها فإن محتوياتها تتجدد وفقاً لآخر تحديث. إن ما حدث ، أن البرنامج التطبيقى قد استقبل الرسالة ، وبمجرد أن أصبح هذا التطبيق هو التطبيق الحالى ، فإنه قام بطلاء نافذته بأحدث المعلومات (لاحظ أن معلومات النافذة المختفية قد تتغير أثناء اختفائها ، ومن اللازم تحديث المعلومات نفسها عند الضغط عليها).

(٤-٣) الرسائل Messages

إن الرسالة تحتوى على حزمة من المعلومات مثل التوقيت ومكان وقع الحدث ، يتم تغليفها فى صورة ماكرو. وهى توجد جميعاً فى الملف Windows.h (أو بأحد ملفاته الفرعية المتضمنة بداخله). وتأخذ

الرسائل أسماء معبرة تبدأ بالحرفين WM (اختصار Windows Message) مثل:

- **WM_LBUTTONDOWN:** رسالة الضغط على الزر الأيمن للفأر.
- **WM_CHAR:** رسالة الضغط على أحد أزرار لوحة الأزرار.
- **WM_PAINT:** رسالة طلاء النافذة بالمحتويات.
- **WM_QUIT:** رسالة إنهاء التطبيق.

وتستمر النوافذ فى إرسال الرسائل المناسبة إلى جميع التطبيقات المفتوحة. ولأن هناك دائما تطبيق واحد نشط (التطبيق الحالى) فإن الرسائل التى ترسل إلى التطبيقات الأخرى غير النشطة (مثل النافذة المختفية فى الخلفية) تختزن فى طابور يسمى طابور الرسائل (Message Queue). ولكل تطبيق طابوره الخاص من الرسائل ، فإذا نشط التطبيق يبدأ فى معالجة الرسائل واحدة تلو الأخرى.

الحلقة التكرارية للرسائل (Message Loop)

يحتوى كل تطبيق نوافذى على حلقة تكرارية لاستلام ومعالجة وتوزيع الرسائل بالصورة الآتية:

```
while (GetMessage (&Msg, NULL, 0, 0)
{
        TranslateMessage(&Msg);
        DispatchMessage(&Msg);
}
```

وتستمر هذه الحلقة التكرارية فى استلام ومعالجة الرسائل طالما أن التطبيق مفتوح حتى تتسلم رسالة إنهاء التطبيق WM_QUIT.

وكما نرى فى شريحة البرنامج السابقة أن هناك ثلاث دوال من دوال الوصلة البينية تتابع باستمرار:

- الدالة **GetMessage()**: لاستقبال الرسائل باستمرار.
- الدالة **TranslateMessage()**: لتفسير الرسالة ، وهى تختص بتفسير ضغطات أزرار اللوحة وتحويلها إلى الأكواد المناسبة.
- الدالة **DispatchMessage()**: لتوزيع الرسالة إلى العنصر المقصود. فالتطبيق الواحد قد يحتوى على عدة نوافذ فرعية.

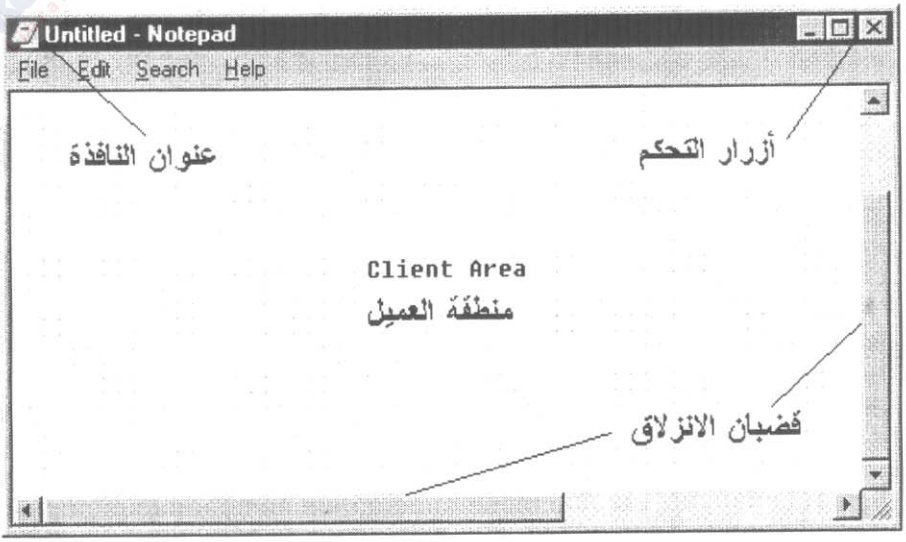
الاستجابة للرسائل (Message Handlers)

إن مهمتك الرئيسية فى كتابة البرنامج النوافذى هو كتابة الكود اللازم للاستجابة للرسائل والذى يصطلح عليه باسم **مقبض الرسالة (Message Handler)**. ويمكنك أن تكتب لكل رسالة من الرسائل المتوقعة المقبض المناسب للاستجابة لها. فعلى سبيل المثال يمكنك أن تحدد ماذا يحدث فى برنامجك عندما يضغط المستخدم على زر الفأر الأيمن كاستجابة للرسالة **WM_LBUTTONDOWN**. ولك مطلق الحرية أن تستجيب لهذا الحدث بالطريقة التى تروق لك ، فربما يؤدى الضغط على هذا الزر إلى ظهور قائمة سريعة (Pop_up menu) بها اختيارات معينة ، أو يؤدى إلى ظهور صندوق رسالة ، أو رسم شكل ما. فماذا يحدث لو أنك لم تكتب المقبض لهذه الرسالة

أو غيرها من الرسائل؟ هل تحدث مشكلة فى بيئة النوافذ؟ بالقطع لا. إذا لم تستجب أنت للرسالة فإن النوافذ تستجيب لها باستخدام مقبض سابق التعريف يسمى **DefWindowProc()**. ولكن النوافذ تبدأ بمقبضك إذا كان جاهزاً ، وإلاّ فإنها تستخدم المقبض سابق التعريف ، وربما لا يؤدى عن هذا المقبض وضيفة ظاهرة ، ولكنه يؤدى إلى استمرار الأحداث فى بيئة النوافذ. هنا نأتى إلى قسم هام من أقسام البرنامج النوافذى وهو الدالة **WindProc()** ، اختصار العبارة (Windows Procedure). وتحتوى هذه الدالة على منشأ شرطى (سويتش) يتضمن كل الرسائل المتوقعة ومقابض الاستجابة لها. وبالطبع فإنه ليس مطلوباً منك أن تستجيب لكل الرسائل ، فهى أكثر من أن تستجيب لها ، إنك تستجيب فقط لما يهمك منها ، وعلى النوافذ أن تستجيب للبقية الباقية.

طلاء النافذة بالمحتويات (WM_PAINT)

يتم طلاء النافذة بمحتوياتها كاستجابة للرسالة WM_PAINT سواء عند بداية إنشاء النافذة أو عند انتهاء صلاحية محتوياتها. وتقوم النوافذ بمهمة إنشاء إطار النافذة وأدوات التحكم الموجودة بها مثل قضبان الانزلاق وأزرار التكبير والتصغير ، أما مهمة المبرمج فهى الإمداد بالمحتويات الداخلية للنافذة التى تقع فى منطقة العميل (Client Area). أنظر الشكل التالى الذى يوضح نافذة البرنامج النوتة ومنطقة العميل به.



شكل (٤-١) منطقة العميل (Client Area)

منطقة العرض (Display Context)

لكل منطقة عميل ، هدف يرتبط بها يسمى **منطقة العرض** (Device Context) ، واختصاره **DC** ، يحتوى على صفحة الكتابة والأدوات اللازمة للكتابة أو الرسم. ويتم التوصل إلى منطقة العرض عن طريق مقبض يسمى **مقبض منطقة العرض HDC**.

المقابض فى بيئة النوافذ (Handles)

فى بيئة النوافذ ، فإن التوصل إلى الأهداف عموماً يتم عن طريق الحصول على مقابضها ولذلك فإن أنماط المقابض (Handles) تعتبر من الأنماط المميزة لبيئة النوافذ. وسوف تلاحظ فى الفقرات التالية أن أغلب دوال الوصلة API تستخدم المقابض كأحد بارامتراتها.

وكفكرة عامة فإن الجدول التالى يوضح أهم مقابض الأهداف التى سنستخدمها فى معالجة النوافذ بهذا الكتاب.

الهدف	اسم المقبض
النافذة (Window)	HWND
قلم الكتابة (Pen)	HPEN
قائمة (Menu)	HMENU
أيقونة (Icon)	HICON
بنط الحروف (Font)	HFONT
مؤشر (Cursor)	HCURSOR
فرشاة رسم (Brush)	HBRUSH
سطح الكتابة أو الرسم (Bitmap)	HBITMAP
جدول التعجيل (Accelerator table)	HACCEL
منطقة العرض (Device Context)	HDC
ملف (File)	HFILE
منطقة (Region)	HRGN

إن كلمة المقبض (Handle) ترتبط أساساً بالأهداف ، وهى بالنسبة للهدف تماثل مقبض الباب الذى تستخدمه فى فتح وقفل الباب. أما مقبض الرسالة (Message Handler) فهو عبارة عن شريحة من الكود تختص بالاستجابة لرسالة ما. ولا يجوز الخلط بين هذا المقبض وسائر المقابض.

(٤-٤) مجرى الأحداث فى البرنامج النوافذى

فيما يلى نلخص "دورة حياة" البرنامج النوافذى ومجرى الأحداث به منذ بدء التشغيل حتى الانتهاء.

١. عندما يبدأ تشغيل البرنامج فإن النوافذ (نظام التشغيل) يستدعى الدالة **WinMain()** وهى الدالة الرئيسية بالبرنامج النوافذى التى تناظر الدالة **main()** فى برامج الكونصول.

٢. من داخل الدالة الرئيسية ، يتم تسجيل (Registering) اسم وخصائص النافذة الرئيسية للتطبيق.

٣. من داخل الدالة الرئيسية ، يتم استدعاء الدالة **CreateWindow()** لخلق نافذة التطبيق.

٤. من داخل الدالة الرئيسية ، يتم استدعاء الدالة **ShowWindow()** لعرض نافذة التطبيق.

٥. من داخل الدالة الرئيسية ، يتم استدعاء الدالة **UpdateWindow()** لطلاء منطقة العميل بنافذة التطبيق بالمحتويات.

٦. تبدأ الحلقة التكرارية للرسائل بداخل الدالة الرئيسية وحتى ينتهى عمل البرنامج. تتولى هذه الحلقة استقبال الرسائل من طابور الرسائل الخاص بالتطبيق ، ومعالجتها ، وتوزيعها على النوافذ الأبناء التابعة للتطبيق.

٧. تتولى النوافذ الأبناء الاستجابة للرسالة إذا توفر المقبض ، أو تتولى النوافذ الاستجابة سابقة التعريف باستخدام الدالة **DefWindowProc** إذا لم يتوفر المقبض لرسالة ما.

٨. ينتهى البرنامج عندما يتلقى الرسالة **WM_QUIT** من النوافذ.

إن هذه هى الأحداث العامة التى تجرى فى أى برنامج نوافذى بصرف النظر عن اللغة المكتوب بها ، ومع ذلك فإن استخدام مؤسسة الفصائل MFC ــ كما سنرى فى الفقرات التالية ــ سوف يوفر على المبرمج الكثير من هذه التفصيلات ، حيث يقتصر عمل المبرمج أساساً فى الاستجابة للرسائل المختلفة.

تذكر هذه المصطلحات

Win32 API	دوال الوصلة البينية API ٣٢ بت
Win16 API	دوال الوصلة البينية API ١٦بت
Wizard	الساحر
MFC	مكتبة مؤسسة الفصائل
MFC Wizard	ساحر مكتبة مؤسسة الفصائل
Messages	الرسائل
Events	الأحداث
Multitasking	المهام المتعددة
Threads	الخيوط
Multithreading	الخيوط المتعددة

Message Queue	طابور الرسائل
Message Handler	مقبض الرسالة
Handle	مقبض (هدف)