ECHO %windir% \ تعرض قيمة متغير البيئة windir (مجلد Windows)

SET \ تعرض جميع متغيرات البيئة المعرّفة في الجلسة الحالية

ECHO %WinDir% \ تعرض قيمة متغير محدد باستخدام علامتَي % من الجانبين

ECHO %Number_Of_Processors% \ تعرض قيمة متغير يحتوي مسافات وحالات مختلطة

ECHO %PATH% \ تعرض قائمة المجلدات التي يبحث فيها النظام عن الأوامر

path = %PATH%;c:\newdir \ في الجلسة الحالية فقط PATH تضيف مجلداً جديداً إلى متغير

ECHO %PATHEXT% \ تعرض قائمة امتدادات الملفات التي ينفّذها النظام تلقائياً

ECHO %TIME% \ تعرض الوقت الحالي (متغير ديناميكي يُحسب لحظة الاستدعاء)

ECHO %DATE% \ تعرض التاريخ الحالي بالصيغة المحددة في إعدادات النظام

ECHO %CD% \ تعرض المجلد الحالي بمساره الكامل (متغير ديناميكي)

ECHO %ERRORLEVEL% \ تعرض رمز الخروج للأمر أو البرنامج السابق تنفيذه

ECHO %RANDOM% \ تعرض رقماً عشوائياً بين 0 و 32767 في كل استدعاء

SET user \ تعرض جميع متغيرات البيئة التي تبدأ بكلمة "user"

SetX \ تُعيّن قيمة متغير بيئة بصفة دائمة (تبقى بعد إغلاق CMD)

Environment.GetLogicalDrives() \ تُعيد مصفوفة بأسماء جميع محركات الأقراص المنطقية المتاحة

string.TrimEnd(' ', ';') \ تحذف المسافات والفواصل من نهاية السلسلة النصية

Environment.MachineName \ تُعيد اسم الحاسوب الحالي على الشبكة

Environment.OSVersion \ تُعيد معلومات إصدار نظام التشغيل الموجود على الجهاز

Environment.CurrentDirectory \ تُعيد مسار المجلد الحالي الذي يعمل فيه التطبيق

Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles) \ تُعيد مسار مجلد Program Files حيث تُنصَّب البرامج

Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFiles) \ تُعيد مسار مجلد الملفات المشتركة داخل Program Files

Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory) \ تُعيد مسار سطح المكتب الخاص بالمستخدم الحالي

Environment.GetFolderPath(Environment.SpecialFolder.Favorites) \ تُعيد مسار مجلد المفضلة لتخزين الروابط المفضلة

Environment.GetFolderPath(Environment.SpecialFolder.History) \ تُعيد مسار مجلد سجل التصفح في نظام Windows

Environment.GetFolderPath(Environment.SpecialFolder.Personal) \ تُعيد مسار مجلد المستندات للمستخدم (My Documents)

Environment.GetFolderPath(Environment.SpecialFolder.Programs) \ تُعيد مسار مجلد البرامج في قائمة ابدأ (Start Menu)

Environment.GetFolderPath(Environment.SpecialFolder.Recent) \ تُعيد مسار مجلد الملفات التي فُتحت مؤخراً

Environment.GetFolderPath(Environment.SpecialFolder.SendTo) \ تُعيد مسار مجلد "إرسال إلى" في قائمة النقر الأيمن

Environment.GetFolderPath(Environment.SpecialFolder.StartMenu) \ تُعيد مسار مجلد قائمة ابدأ الرئيسية

Environment.GetFolderPath(Environment.SpecialFolder.Startup) \ تُعيد مسار مجلد البرامج التي تبدأ تلقائياً مع تشغيل Windows

Environment.GetFolderPath(Environment.SpecialFolder.System) \ تُعيد مسار مجلد System في Windows (عادةً C:\Windows\system32)

Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) \ تُعيد مسار مجلد بيانات التطبيق الخاص بالمستخدم (Roaming)

Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData) \ تُعيد مسار مجلد بيانات التطبيقات المشترك بين جميع المستخدمين

Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData) \ تُعيد مسار مجلد بيانات التطبيق المحلية للمستخدم (Local)

Environment.GetFolderPath(Environment.SpecialFolder.Cookies) \ تُعيد مسار المجلد المخصص لتخزين ملفات الكوكيز

getenv_s(&size, NULL, 0, "path") \ تحسب الحجم المطلوب لتخزين قيمة متغير PATH دون استرجاعها بعد

getenv_s(&size, NULL, 0, "pathext") \ تحسب الحجم المطلوب لتخزين قيمة متغير PATHEXT

getenv_s(&size, NULL, 0, "programfiles") \ تحسب الحجم المطلوب لتخزين قيمة متغير PROGRAMFILES

getenv_s(&size, NULL, 0, "appdata") \ تحسب الحجم المطلوب لتخزين قيمة متغير APPDATA

malloc(size * sizeof(char)) \ تحجز مساحة ذاكرة كافية لاستقبال قيمة متغير البيئة

getenv_s(&size, path_p, size, "path") \ تسترجع القيمة الفعلية لمتغير PATH وتضعها في المتغير المخصص

getenv_s(&size, pathext_p, size, "pathext") \ تسترجع القيمة الفعلية لمتغير PATHEXT

getenv_s(&size, programfiles_p, size, "programfiles") \ تسترجع القيمة الفعلية لمتغير PROGRAMFILES

getenv_s(&size, appdata_p, size, "appdata") \ تسترجع القيمة الفعلية لمتغير APPDATA

cout << path_p \ تطبع قيمة PATH على الشاشة

system("pause") \ توقف تنفيذ البرنامج وتنتظر ضغط المستخدم على أي مفتاح

string[] args \ مصفوفة تستقبل الوسيطات الممررة للبرنامج من سطر الأوامر

args.Length \ تُعيد عدد الوسيطات التي مُررت للبرنامج عند تشغيله

args[i] \ تُعيد قيمة الوسيطة عند الفهرس i من سطر الأوامر

Console.WriteLine("args[{0}] = {1}", i, args[i]) \ تطبع رقم وقيمة كل وسيطة بشكل منسق على الشاشة

Convert.ToInt32(args[i]) \ تحوّل الوسيطة النصية من سطر الأوامر إلى عدد صحيح

sum += e \ تجمع قيمة الوسيطة الحالية إلى المتراكم للحصول على مجموع الوسيطات

cmdline1.exe A B C \ تشغيل البرنامج مع تمرير ثلاث وسيطات نصية له

sumArgs.exe 1 2 3 4 5 6 7 8 100 500 \ تشغيل برنامج الجمع مع تمرير عشرة أرقام كوسيطات

csc cmdline1.cs \ تصريف (compile) ملف C# من سطر أوامر Visual Studio

csc sumArgs.cs \ تصريف ملف برنامج جمع الوسيطات C#

إدارة النظام :C — سلايد 7

system(command); \ تنفيذ أمر نظام تشغيل من داخل البرنامج — يتطلب #include <cstdlib> getenv_s \ قراءة قيمة متغير بيئة النظام بأمان — يتطلب #include <stdlib.h>

إدارة الملفات :C# — سلايد 7 (System.IO)

File.Create("path") \ إنشاء ملف جديد في المسار المحدد File.Copy(src, dst) \ نسخ ملف من مسار مصدر إلى
File.Exists("path") \ التحقق من وجود الملف وإعادة true أو false File.GetAttributes("path") \ مسار وجهة
ReadOnly File.SetAttributes("path", الحصول على خصائص الملف مثل Hidden أو
FileAttributes.ReadOnly) \ تعيين خاصية للملف مثل جعله للقراءة فقط Directory.Exists("path") \ التحقق
Directory.GetParent("path") \ الحصول على المجلد الأب للمسار المحدد من وجود المجلد في المسار المحدد
Directory.Delete("path", true) \ حذف مجلد مع جميع محتوياته (true = حذف متكرر)
Directory.GetFiles(@"D:") \ إعادة قائمة بأسماء جميع الملفات داخل المجلد Path.Combine(sourcePath,
fName) \ دمج جزأين من مسار ملف بصورة صحيحة ومتوافقة مع النظام

سلايد 7 — C#: البيئة والعمليات (System / System.Diagnostics)

using الحصول على مسار مجلد نظامي خاص — يتطلب Environment.GetFolderPath(...) \
Program تعداد يمثل مسار مجلد Environment.SpecialFolder.ProgramFiles \ System.Collections
Files الحصول على مصفوفة بجميع العمليات الجارية حالياً في النظام Process.GetProcesses() \ في النظام

سلايد 8 — C#: العمليات والخيوط (System.Diagnostics / System.Threading)

PID الحصول على عملية معينة باستخدام معرّفها الرقمي Process.GetProcessById(1234) \
الحصول على جميع العمليات الجارية التي تحمل الاسم المحدد Process.GetProcessesByName("notepad") \
Process.Start("IExplore.exe", تشغيل تطبيق خارجي من داخل البرنامج Process.Start("IExplore.exe") \
"www.ypu.edu.sy") \ تشغيل تطبيق مع تمرير وسيط له (مثل رابط للمتصفح) Thread t1 = new
Thread(b.RunMe1); \ إنشاء خيط تنفيذ جديد يُنفّذ الدالة RunMe1 — يتطلب using System.Threading
Thread t2 = new Thread(b.RunMe2); \ إنشاء خيط تنفيذ ثانٍ مستقل يُنفّذ الدالة RunMe2

ملاحظة: السلايدات 1 إلى 6 نظرية بحتة — لا تحتوي على أي تعليمات برمجية قابلة للتنفيذ.

هذا شرح خصائص ودوال الـ Process في C#:

p.Id \ يُرجع المعرّف الفريد (PID) للعملية

p.ProcessName \ يُرجع اسم العملية الجارية

p.StartTime \ يُرجع وقت بدء تشغيل العملية

p.TotalProcessorTime \ يُرجع إجمالي وقت المعالج الذي استهلكته العملية

p.StartInfo \ يُحدد خصائص بدء التشغيل التي تُمرَّر إلى دالة Start

p.Threads \ يُرجع مجموعة الخيوط (Threads) التي تعمل داخل العملية

p.Threads.Count \ يُرجع عدد الخيوط الموجودة داخل العملية

p.BasePriority \ يُرجع الأولوية الأساسية للعملية

p.MachineName \ يُرجع اسم الجهاز الذي تعمل عليه العملية

Process.GetProcesses() \ تُرجع مصفوفة تحتوي على جميع العمليات الجارية على الجهاز المحلي

Process.GetProcessById(int id) \ تُنشئ كائن Process مرتبطاً بالعملية التي تحمل المعرّف المحدد

Process.GetProcessesByName(string name) \ تُرجع مصفوفة بجميع العمليات التي تحمل الاسم المحدد

Process.GetCurrentProcess() \ تُرجع كائن Process يمثّل العملية الحالية النشطة

Process.Start(string fileName) \ تُشغّل تطبيقاً أو ملفاً بتحديد اسمه فقط

Process.Start(string fileName, string arguments) \ تُشغّل تطبيقاً مع تمرير وسيطات سطر الأوامر إليه

Process.Start(ProcessStartInfo) \ تُشغّل تطبيقاً باستخدام كائن إعدادات متقدمة يحتوي على WindowStyle والوسيطات

p.Kill() \ تُنهي العملية المرتبطة فوراً

p.Close() \ تُحرّر جميع الموارد المرتبطة بكائن العملية

ss.WindowStyle = ProcessWindowStyle.Minimized \ يُحدد حالة نافذة التطبيق عند فتحه (مصغّرة، عادية، مكبّرة...)

ss.Arguments = "www.yahoo.com" \ يُحدد الوسيطات التي تُمرَّر إلى التطبيق عند بدء تشغيله

Process[] list = Process.GetProcesses(); \ استرجاع قائمة بجميع العمليات الجارية وتخزينها في مصفوفة

Process localById = Process.GetProcessById(1234); \ الحصول على عملية بعينها عن طريق رقم معرّفها

Process[] localByName = Process.GetProcessesByName("notepad"); \ الحصول على جميع نسخ Notepad الجارية حالياً

Process.Start("IExplore.exe"); \ تشغيل Internet Explorer بدون وسيطات

Process.Start("notepad.exe", @"D:\file1.txt"); \ تشغيل Notepad مباشرةً مع فتح ملف نصي محدد

dataGridView1.Rows.Add(p.Id, p.ProcessName, p.Threads.Count); \ إضافة بيانات العملية إلى جدول واجهة رسومية (ID، الاسم، عدد الخيوط)

تعليمات الـ Threads والـ Locking — C# (حتى الشريحة 14)

المكتبات — using

using System.Threading; \ تضمين المكتبة الأساسية للطباعة والإدخال والإخراج using System; \ تضمين الضرورية لإنشاء وإدارة الخيوط Threads مكتبة الـ

إنشاء وتشغيل Thread

new Thread(method) \ إنشاء خيط جديد مرتبط بالدالة المحددة t.Start() \ بدء تنفيذ الخيط وتشغيله بالتوازي مع بقية الخيوط Thread.CurrentThread \ الحصول على مرجع الخيط الحالي الذي ينفذ الكود Name \ تعيين اسم للخيط أو قراءته لأغراض التعرف والتتبع Thread.Sleep(ms) \ إيقاف الخيط الحالي مؤقتًا لعدد معين من الميلي ثانية

مصفوفة الخيوط

Thread[] threads = new Thread[N] \ إنشاء مصفوفة ديناميكية لتخزين عدد N من الخيوط string.Format("...", i) \ بناء نص منسق (كاسم الخيط) من قيم متغيرة foreach (Thread t in threads) \ التكرار عبر جميع الخيوط في المصفوفة لتشغيلها

---

lock — التزامن

lock(this) \ قفل الكائن الحالي لمنع دخول أكثر من خيط واحد للمنطقة الحرجة في نفس الوقت

معالجة الاستثناءات

try { ... } \ تحديد منطقة الكود التي قد ترمي استثناءً لمراقبتها throw new Exception("msg") \ رمي استثناء جديد
catch (Exception ex) { ... } \ التقاط الاستثناء ومعالجته عند وقوعه ex.Message \ قراءة نص رسالة الخطأ المحتوى داخل الاستثناء يحمل رسالة خطأ محددة

الإدخال والإخراج

Console.Write(...) \ طباعة نص Console.WriteLine(...) \ طباعة نص متبوع بسطر جديد إلى نافذة الأوامر
Console.ReadKey() \ إيقاف البرنامج وانتظار ضغط أي مفتاح من المستخدم بدون سطر جديد (يكمل بجانب السابق)
Console.ReadLine() \ إيقاف البرنامج وانتظار إدخال سطر كامل من المستخدم

العمليات الرياضية والعشوائية

new Random() \ إنشاء مولّد أرقام عشوائية r.Next(min, max) \ توليد رقم عشوائي بين الحد الأدنى والحد الأقصى

الحلقات والتحكم

for (int i = 0; i < N; i++) \ حلقة تكرار بعدد محدد مسبقًا من المرات return amount; \ إنهاء الدالة وإعادة قيمة
return 0; \ إعادة صفر للدلالة على رفض العملية أو فشلها المبلغ إلى المستدعي

المفاهيم الخاصة بالخيوط

new Thread(acc.DoTransactions) \ ربط الخيط بدالة كائن موجود (instance method) لا بدالة ثابتة
Thread.CurrentThread.Name \ قراءة اسم الخيط الحالي أثناء تنفيذه لعرضه أو تسجيله