PROGRAMAR

A REVISTA PORTUGUESA DE PROGRAMAÇÃO

Revista nº8 - Maio de 2007

www.portugal-a-programar.org

Windows Vista para programadores

Conheça as mudanças no novo sistema operativo da Microsoft que vão afectar o desenvolvimento das suas aplicações.

Programação em LUA

Descubra esta linguagem de programação extremamente poderosa.

Microcontroladores

Protocolos SPI

Protocolo SPI aplicado à comunicação entre microcontroladores e periféricos com interface em Visual Basic.

Ubuntu 7.04 Feisty Fawn

e ainda...

Segurança em Sistemas Distribuídos

índice

equipa PROGRAMAR

administração
Rui Maia
David Pintassilgo

coordenador
Sérgio Santos

coordenador adjunto
Miguel Pais

editor
Joel Ramos

redacção
Fábio Pedrosa
Bruno Vaz
Fábio Correia
Pedro Teixeira
Ricardo Rocha
Sandro Pinto
Miguel Wahnan
Marco Silva
José Oliveira

colaboradores
Rui Gonçalves
Daniel Correia

contacto
revistaprogramar
@portugal-a-programar.org

website
www.revista-programar.info

A Equipa

É dito que, para além de ter uma boa ideia, ter uma boa equipa é essencial para obter sucesso num projecto. Começar e tentar acabar um projecto sozinho é, muitas vezes, suicídio para o próprio projecto. Mesmo que as nossas capacidades estejam à altura, é sempre necessário outros pontos de vista, opiniões, outras soluções para um determinado problema e, principalmente, apoio e motivação. A probabilidade de um projecto ficar a meio, quer por desinteresse ou falta de tempo, é muito maior quando estamos sozinhos à volta dele.

Muitas vezes a ideia nem é o principal já que nem sempre a ideia que começamos é a final. Existem muitos casos de sucesso actualmente que demonstram isso mesmo. Ainda mais, uma boa equipa consegue sempre criar boas ideias quando é preciso.

O tamanho das equipas também é muitas vezes discutido. Na minha opinião, para começar do zero o melhor é mesmo começar com um grupo reduzido. Todos os factores "extra-projecto" como a comunicação e organização aumentam exponencialmente com o número de pessoas envolvidas no projecto. Mas irá chegar o tempo em que serão precisas mais pessoas. Mas, em vez de atirar mais gente para um problema, muitas vezes é mais útil dividir a equipa em vários grupos com diferentes áreas de foco. E o principal: respeitar sempre o trabalho e as opiniões de todos os membros da equipa.

Sérgio Santos

Dell e Ubuntu

Depois de uma votação no site da Dell dedicado aos consumidores, o IdeaStorm, a Dell decidiu começar a comercializar computadores com o Ubuntu 7.04 (Feisty Fawn) como sistema operativo pré-instalado. A escolha recaiu sobre o Ubuntu devido à maioria dos consumidores o ter preferido. A Dell não irá distribuir a variante KDE, o Kubuntu, apenas o Ubuntu que usa GNOME.

A votação no IdeaStorm envolveu mais de 100 mil pessoas e revelou que cerca de 70% destas estavam dispostas a utilizar sistemas operativos Linux.

Numa fase inicial estes computadores apenas estarão disponíveis para venda nos EUA, mas a Dell não rejeita a hipótese de expandir esta iniciativa a outros países.

Gaim muda nome para Pidgin

Depois dum longo e conflituoso processo de negociações com a AOL, um dos principais ISPs dos EUA, a equipa do Gaim decidiu mudar o nome do famoso Instant Messenger para Pidgin. Em causa estava o sistema de comunicação em tempo real AIM, detido e registado pela AOL, pouco tempo depois das primeiras versões do Gaim terem vindo a público.

Após muito tempo em tribunal, a equipa do Gaim decidiu dar o braço a torcer uma vez que o processo estava a impedir o lançamento de versões non-beta do Gaim.

Assim, foi possível a tão aguardada versão 2.0.0. A nova versão tem um novo visual, muitas novas funcionalidades e claro, um novo nome: Pidgin.

Google é a marca mais valiosa do mundo

Segundo um estudo da empresa de pesquisa de mercado Millward Brown em conjunto com o Financial Times, o motor de busca Google é a marca mais cara do mundo estando avaliada em qualquer coisa como 49 mil milhões de euros. De acordo com o ranking BrandZ - As 100 Marcas Mais Poderosas, o Google aumentou 77% num ano, tendo ultrupassado a Microsoft, a mais valiosa em 2006.

Este ano, a Microsoft passou para terceiro lugar valendo 40,5 mil milhões de euros, estando abaixo da General Electrics que vale 45,6 mil milhões de euros. O quarto posto é ocupado pela Coca-Cola, com 32,5 mil milhões de euros, um pouco aquém do valor da Microsoft.

Entre as 10 marcas com valores mais elevados, oito são norte- americanas, sendo as de tecnologias e as de consumo geral as mais presentes na lista.

No conjunto, estas 10 marcas - Google, Microsoft, China Mobile, Wal-Mart, Coca-Cola, Citi, IBM, Toyota, McDonald's e Bank of América - valem 1,2 biliões de euros, mais 10,6% do que no ano passado.

O segmento de tecnologias é aquele onde as marcas valem mais, sendo que as cinco mais poderosas (Google, Microsoft, IBM, Nokia e HP) chegam aos 155,6 mil milhões de euros.

Windows Vista para Programadores

Introdução

Windows Vista (antigo *codename*: Longhorn) é o mais recente lançamento da Microsoft, mais de 5 anos após o seu sucessor Windows XP, contém centenas de novas características, entre as quais, uma nova interface gráfica, melhorias de pesquisa, ferramentas multimédia, áudio, segurança, … e principalmente no assunto que nos interessa (plataforma de desenvolvimento), a nova versão da plataforma .NET.

Juntamente com o lançamento do sistema, vem também integrada a nova versão da Plataforma .NET 3.0 (antigo codename: WinFX).

Baseada na versão 2.0, a nova versão da plataforma traz novas APIs, como:

• Windows Presentation Foundation (antigo Avalon) simplifica a construção de aplicações cliente ricas, uma vez que facilita a integração de multimédia, documentos e UI. O WPF introduz um sistema avançado de layouts, fácil integração de multimédia e gráficos vectoriais e 3D, facilitando ainda a colaboração entre programadores e designers. Juntamente com esta tecnologia, a Microsoft introduziu uma nova linguagem de marcação baseada em XML, chamada XAML.

• Windows Communication Foundation (WCF) fornece uma Framework única para a construção rápida de aplicações seguras, fiáveis e orientadas a serviços.

• Windows Workflow Foundation (WWF) oferece tarefas de automação e transacções integradas utilizando gráficos de fluxo. É um modelo de programação, motor e ferramentas para permitir aplicações baseadas em fluxo em Windows.

• Windows CardSpace (WCS) é um componente que guarda identidades digitais duma pessoa em segurança, e oferece uma interface unificada para escolher uma identidade para uma particular transacção, tal como o login de um website.

API, de Application Programming Interface (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para utilização das suas funcionalidades por programas aplicativos, isto é: programas que não querem envolver-se em detalhes da implementação do software, mas apenas usar os seus serviços.

Estas tecnologias também estão disponíveis para Windows XP e Windows Server 2003.
Existem também novas APIs importantes no núcleo do sistema operativo, inclusive o renovado interface áudio, rede, impressão e vídeo. Alterações na infra-estrutura de segurança, melhoramentos na distribuição e instalação de aplicação ("ClickOnce" e Windows Installer 4.0), novo modelo de desenvolvimento de drivers (Windows Driver Foundation), Transaction NTFS, desenvolvimento na API de computação móvel (gestão de energia, Suporte Tablet PC Ink, SideShow) e grandes desenvolvimentos dos subsistemas como Winlogon e CAPI.

.NET 3.0 e os novos subsistemas

Inicialmente o Longhorn, agora chamado Windows Vista, era um projecto ainda mais ambicioso onde a nova Framework iria substituir a actual API base dos sistemas operativos Windows (Win32) e alterava o ambiente de desenvolvimento em Windows para sempre, utilizando código gerido (managed code). De facto esta ideia foi abandonada algures durante o desenvolvimento do sistema, mas no entanto esta nova Framework continua a ser uma grande mudança.

Porquê uma nova API?

Parte da razão, é sem dúvida a velocidade a que as tecnologias Hardware estão a ser desenvolvidas. Os computadores pessoais já começam a utilizar discos rígidos com capacidades enormes, monitores de alta-resolução, velocidades de processamento enormes, etc. Conectividade é também de grande interesse, juntamente com a necessidade de melhor segurança. Standards como o RSS e Serviços Web estão a ganhar adopção e a resolver problemas de integração. Outra razão é o lema "Write once, run everywhere", bastante utilizada para descrever esta plataforma, visto que a principal falha que a API Win32 (mais de 12 anos de existência) acarretava era a necessidade de manter compatibilidade entre alterações sem afectar as aplicações que a utilizavam.

Com esta "nova" Framework este problema desaparece, pois é garantida 100% compatibilidade. Até à versão 2.0 da Framework, qualquer aplicação desenvolvida utilizando a plataforma corria estável e perfeitamente em sistemas Windows 98 e superiores. Com a nova versão (3.0) apenas o Windows XP SP2 e superiores a suportam. Existem outras melhorias na utilização desta Framework, mas não é do âmbito deste artigo.

Win32 vs. NET

Como é habitual nos sistemas operativos, um grande esforço é feito para manter compatibilidade nas aplicações, como a execução de aplicações Win32 em Windows Vista. Algumas aplicações podem não tirar vantagem das novas APIs incluídas nas novas versões da Framework, mas devem continuar a trabalhar igualmente às versões anteriores do SO.Por outro lado, a Microsoft garante continuar a adicionar APIs geridas (managed APIs) à Framework, o que pode não acontecer na Win32.

Cabe aos programadores escolherem entre continuar a utilizar a API antiga, adaptar as aplicações para utilizar as novas APIs geridas (managed APIs) utilizando código de adaptação ou "simplesmente" reescrever a aplicação utilizando código gerido (managed code).Como é óbvio existem grandes melhorias de produtividade, simplicidade e compatibilidade em fazer a passagem para código gerido (managed code).

Outra das vantagens na utilização de código gerido é o benefício de poder utilizar várias linguagens, que no caso da Framework da Microsoft actualmente corresponde ao C++, C#, VB.NET, ASP.NET e J#. Com novas versões da Framework são implementadas novas alterações nas linguagens, tal como outras ferramentas. É também possível utilizar outras linguagens com a Framework, como o IronPython (implementação do Python) e Ruby .NET (implementação do Ruby).

Windows Presentation Foundation

Como falado anteriormente, Windows Presentation Foundation (WPF) é o nome de um conjunto novo de API's na nova versão da Plataforma .NET que permite a qualquer programador dar uso ao novo modelo gráfico que utiliza melhor do que nunca o poder gráfico do processador gráfico (GPU).

Qualquer aplicação construída utilizando WPF constitui tipicamente de um número de páginas Extensible Application Markup Language (XAML) juntamente com o seu código suporte. Obviamente a funcionalidade que uma colecção de páginas independentes pode oferecer é limitada, e nâo é suficiente para grande parte das necessidades. Por exemplo, o processamento que ocorre ao nível da página não suporta necessidades como guardar o estado da página enquando o utilizador navega, muda de página, carrega páginas, gere variáveis globais, etc.

WPF ou Windows Form?
WPF é claramente para aplicações com uso multimédia e layout avançado. A não ser que o suporte a Windows 98 seja mesmo necessário, então o WPF é um excelente substituto para o "antigo" WinForms.

Como tal, o modelo da aplicação WPF suporta a colecção de páginas XAML em uma única aplicação do modo tradicional.

Em termos de aplicações Windows Forms, em que uma página XAML é semelhante a um ficheiro de formulário (WinForm), a aplicação Windows Forms reune os formulários num único executável do mesmo modo que uma aplicação WPF reune as suas páginas XAML.

Muitos pensam que o WPF depende do XAML, mas isto não é verdade. Aliás, tudo o que se constrói utilizando XAML pode ser feito em qualquer linguagem .NET. Até porque o XAML não é apenas utilizado pelo WPF, é também por outras tecnologias (Windows Workflow Foundation [WWF] por exemplo). Aplicações WPF podem correr em duas diferentes formas:

• Aplicações de browser XAML: Elas correm como aplicações alojadas no browser. Elas não correm no sistema do utilizador e não correm offline. Do ponto de vista da segurança, elas executam na zona de segurança de Internet, que limita o seu acesso a recursos do sistema. Visto que correm com um alto nível de segurança, não necessitam de permissão do utilizador para correr.

• Aplicações instaladas: Nesta forma, as aplicações ou são alojadas numa janela (como qualquer aplicações Windows) ou alojadas na janela de navegação WPF. Esta janela contém uma banda de navegação na região do cliente com botões de navegação. Esta banda é substituível pelos programadores que queiram construir a sua própria barra utilizando a classe System.Windows.Navigation.NavigationWindows.

Para cada um destes dois tipos de aplicação e três de distribuição, com a escolha do método a utilizar, existe impacto na segurança. Como tal, convém ao programador conhecer concretamente cada tipo.

XAML

XAML (pronuncia-se "zamel") é um dos principais conceitos associados com o WPF, pois fornece ao designer e programador uma forma concisa de representar uma interface gráfica ou outras hierarquias de objectos. XAML é uma linguagem declarativa de marcação baseada em XML.

Muitos pensam que o WPF depende do XAML, mas isto não é verdade. Aliás, tudo o que se constrói utilizando XAML pode ser feito em qualquer linguagem .NET. Até porque o XAML não é apenas utilizado pelo WPF, é também por outras tecnologias (Windows Workflow Foundation [WWF] por exemplo).

Das inúmeras vantagens em utilizar estas páginas XAML, que basicamente são ficheiros XML, uma das principais vantagens deste processo é a separação entre o design do interface gráfico e o código de fundo. Esta separação ajuda consideralmente quando a equipa ou pessoa que faz o design não é a mesma que a associada com a programação.Por exemplo, a seguinte imagem mostra a diferença entre C#/VB.NET/XAML na construção de um simples botão:

Existem muitas ferramentas para execução e criação de Xaml. Juntamente com o Visual Studio 2005 vêm o XamlPad que permite introduzir código Xaml e ver o resultado imediato. Pode também apenas guardar o código num ficheiro de extensão .xaml e abrir no Internet Explorer (necessita do .NET 3.0 instalado).

Como o WPF introduz tantos novos conceitos, não é prático desenvolver código e design sem novas ferramentas apropriadas. Desde o lançamento do Visual Studio 2005 que este juntamente com a extensão para .NET 3.0 oferece um bom ambiente para construir aplicações WPF, com novos tipos de projectos e editores visuais específicos para XAML.
Com a nova versão Visual Studio "Orcas" (ainda em desenvolvimento) este suporte ainda será ambrangido.

Repare que por exemplo, o código Xaml não necessita do LightBlue declarado como membro de Colors, é reconhecido automaticamente. Utilizando este exemplo dado, podemos também mostrar a declaração de eventos.

Ao adicionar o atributo Click="button_click" na tag Button no código XAML seria o equivalente a adicionar um evento, que executaria o método button_click sempre que o utilizador clicasse no botão criado. Esta adição, implicaria a necessidade de juntar o XAML com uma linguagem procedimental .NET.

Existem ainda outras ferramentas profissionais para criação de conteúdo WPF, algumas destas também da Microsoft.

Microsoft Expression Blend é uma das principais ferramentas utilizadas para estes formatos, principalmente porque partilha o mesmo sistema de ficheiros que os projectos do Visual Studio, facilitando o trabalho em simultâneo no código e design.

Microsoft Expression Data também é outra ferramenta Microsoft que permite criação de imagens bitmap e vectoriais com exportação para XAML.

Silverlight

Conhecido por WPF/E (Windows Presentation Foundation/Everywhere), Silverlight é uma versão gratuita e reduzida da plataforma WPF compatível com vários browsers e vários sistemas operativos, garantindo assim a compatibilidade de aplicações WPF para Browser em grande parte dos sistemas existentes. Actualmente já suporta Windows, Linux e Mac.

Em competição directa ao Flash player da Adobe, este plug-in traz o mundo multimédia do WPF ao browser.

Windows Communication Foundation

WCF é o núcleo das aplicações com comunicações orientadas a serviços para a plataforma Windows. Está construído no topo dos standards de serviços web e está desenhado para as necessidades do service-oriented architecture (SOA) e software as a service (SAAS). A filosofia por detrás destes é que o software deve ser escrito utilizando interfaces que comunicam utilizando protocolos standard.

Serviços web foram o ínicio desta história, onde lógica dos negócios é implementada em puro texto, baseado em XML, utilizando uma linguagem chamada Web Service Description Language (WSDL). Programadores podem então utilizar uma mensagem num formato chamado Simple Object Access Protocol (SOAP) que utiliza o XML, para comunicar com o serviço. No entanto serviços web estão limitados e não suportam algumas funcionalidades como segurança, rentabilidade, e transacções sem complicadas adições no mínimo.

WCF está desenhado para permitir tudo isto e muito mais; oferece uma API que permite implementar serviços sem preocupação com estes detalhes, libertando tempo útil para melhor concentração na lógica do negócio.

WCF oferece o modelo orientado a serviço, construído no topo da plataforma .NET e unifica as diferentes capacidades de sistemas distribuídos, incluindo serviços web estáticos, mensagens, transacções e remoting numa única API.

WCF oferece uma nova namespace, chamada System.ServiceModel, que fornece as ferramentas para alguns dos cenários utilizados em aplicações distribuídas:

- Mensagens, num único sentido ou ambos
- Chamadas a procedimentos remotos
- Callbacks
- Sessões
- Segurança
- Transacções
- etc

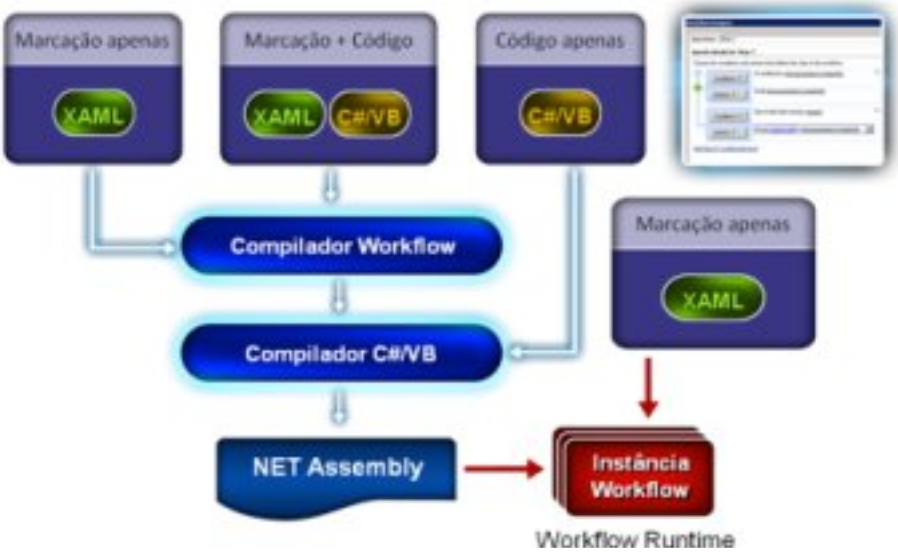
WCF é uma enorme e importante API, o leitor se interessado já pode encontrar alguns livros que cobrem o assunto (nenhum em português actualmente).

Windows Workflow Foundation

WWF é mais uma das novas tecnologias incluída no .NET 3.0 e permite definir, executar e gerir fluxos de trabalho. Utilizando um fluxo de trabalho podemos declarar um programa como uma série de passos compostos de actividades. Estas actividades podem ser facilmente criadas por código, marcação ou através de outras actividades criadas.

Existem inúmeras vantagens em utilizar estes fluxos de trabalho ao invés de escrever código. Em primeiro lugar, para programadores, fluxos de trabalho são mais fáceis de entender do que código, porque fluxos de trabalho oferecem uma representação do processo. Consequentemente, facilita o trabalho quando o programador precisa de modificar o fluxo associado a um processo, que geralmente não é nada fácil para o programador modificar no código.

Semelhante ao WPF, o WWF também utiliza marcação XAML para declarar a estrutura do fluxo de trabalho. Contudo, esta declaração também pode ser feita por código de uma das linguagens .NET.
Mais informação sobre esta tecnologia pode ser encontrada no site msdn.microsoft.com/workflow.



Windows CardSpace

Hoje, a maior parte dos websites como o Hotmail, eBay, Amazon, Paypal utilizam uma combinação de nome e palavra-chave para garantir acesso a informações pessoais associadas a uma conta. Existem dois problemas fundamentais com este método de segurança. Primeiro, utilizadores são obrigados a criar e relembrar uma lista sempre a crescer de nomes/palavras-chave (tipicamente uma combinação por cada website que visita). Isto geralmente gera práticas inseguras, como utilizar o mesmo nome e palavra-chave para vários websites, ou guardar listas em texto inseguro (desencriptado) no computador ou papel.

Segundo, nomes de utilizador e palavras-chave são susceptíveis a uma ameaça em crescimento de ataques de fraude de várias formas, incluído "phishing". Ataques Phishing enganam os utilizadores pedindo para fornecer as informações de identificação a websites fraudulentos. Levam a vantagem da incapacidade do utilizador confirmar a validade do website que visitam.

O que é o CardSpace?

CardSpace é o nome de uma nova tecnologia do Windows que simplifica e melhora a segurança de acesso a recursos e partilha de informação pessoal na Internet. CardSpace ajuda a gerir a informação pessoal e a controlar a quem essa informação é dada, reduzindo o número de problemas associados ao método tradicional de utilizador/palavra-chave.

Em vez de forçar os utilizadores a lembrar uma quantidade enorme de utilizadores e palavras-chave, o CardSpace fornece aos utilizadores um método mais seguro equivalente aos utilizadores e palavras-chave, utilizando um conjunto de cartões de identificação virtual que podem ser acedidos por um interface. Estes cartões podem ser utilizados para fazer login apenas com um clique em websites com compatibilidade à tecnologia.

Conclusão

A Microsoft e os seus recentes lançamentos vieram novamente subir a fasquia na programação em ambientes Windows. Depois de uma pequena e simples visita guiada pelas principais introduções na plataforma .NET 3.0, cabe agora ao leitor/programador preparar-se para os novos desafios que com isto vão chegar.

Fábio Pedrosa

Estatísticas em PHP 2ª Parte

Depois de uma primeira parte na edição anterior desta revista, o artigo sobre a criação de estatísticas em PHP será concluído nesta edição.

Tabelas Resumo

A próxima função irá gerar uma tabela de 3 colunas com os respectivos dados resumo preenchidos. No entanto existe mais 3 tabelas resumo no Script Completo que permite gerar resumos com informação diferente.

```
function getCountrys ( $beginDate , $endDate ) {
  include ( "connection.php" );
  mysql_select_db ( $database , $connection );
  $sql_countrys = "SELECT cod_pais, nome_pais, count(nome_pais) as total
                   FROM estatisticas WHERE data >=
                   ".changeDate($beginDate)."' AND data <=
                   '".changeDate($endDate)."' GROUP BY cod_pais
                   ORDER BY total DESC";

  $countrys = mysql_query($sql_countrys,$connection) or die(mysql_error());

  while($row_countrys = mysql_fetch_array ($countrys , MYSQL_ASSOC ) ) {
    $total_hits += $row_countrys[ 'total' ];
  }

  $countrys = mysql_query($sql_countrys,$connection) or die(mysql_error());


  $table = "<table width='390' border='0' cellspacing='0' cellpadding='0'
                               align='center'>
            <tr bgcolor='#E4E4EC' class='big_titles'>
              <td colspan='4' align='center'>
                Ranking de Visitas por Pa&#237;s
              </td>
             </tr>
             <tr bgcolor='#E4E4EC' class='item' align='center' valign='middle'>
               <td width='75' height='45'>
                 C&#243;digo Pa&#237;s
               </td>
               <td width='125'>
                 Nome Pa&#237;s
               </td>
               <td width='100'>
                 Total P&#225;ginas Visitadas
               </td>
```

```
              <td width='90'>
                % (Relev&#226;ncia)
              </td>
            </tr>";


  while($row_countrys = mysql_fetch_array($countrys , MYSQL_ASSOC ) ) {

    $table .= "<tr bgcolor='#F8F8F8' class='general_text'>
              <td height='35' align='left'>
                ".$row_countrys[ 'cod_pais' ]."
              </td>
              <td align='left'>
                ".$row_countrys[ 'nome_pais' ]."
              </td>
              <td align='center'>
                ".$row_countrys[ 'total' ]."
              </td>
              <td align='center'>
                ".(round((($row_countrys['total'] / $total_hits)*100),2))." %
              </td>
            </tr>";
  }


  $table .= "<tr height='25' bgcolor='#E4E4EC' class='item'>
              <td>
                 
              </td>
              <td align='right'>
                Total:
              </td>
              <td align='center'>
                ".$total_hits."
              </td>
              <td>
                 
              </td>
            </tr>
          </table>";


  return $table;

} // FIM DE "getCountrys"
```

Desta forma termina-se a Classe, embora o ficheiro 'statistics.php' não esteja finalizado, já que vai ser acedido directamente pela Classe Charts para a geração dos gráficos. Haverá que preparar o ficheiro para essas operações.

Seguidamente à Classe será introduzido o seguinte código:

```
if ( isset ( $_REQUEST[ 'graph' ] ) && $_REQUEST[ 'graph' ] == 1 ) { //
  include ( "charts.php" );
  $statistics = new Statistics;
  SendChartData ( $statistics->getPageViewsGraph ( $_REQUEST[ 'begin_date' ] ,
                                                   $_REQUEST[ 'end_date' ] ) );
}
elseif ( isset ( $_REQUEST[ 'graph' ] ) && $_REQUEST[ 'graph' ] == 2 ) {
  include ( "charts.php" );
  $statistics = new Statistics;
  SendChartData ( $statistics->getVisitGraph ( $_REQUEST[ 'begin_date' ] ,
                                               $_REQUEST[ 'end_date' ] ) );
}
elseif ( isset ( $_REQUEST[ 'graph' ] ) && $_REQUEST[ 'graph' ] == 3 ) {
  include ( "charts.php" );
  $statistics = new Statistics;
  SendChartData ( $statistics->getBrowserGraph ( $_REQUEST[ 'begin_date' ] ,
                                                 $_REQUEST[ 'end_date' ] ) );
}
elseif ( isset ( $_REQUEST[ 'graph' ] ) && $_REQUEST[ 'graph' ] == 4 ) {
  include ( "charts.php" );
  $statistics = new Statistics;
  SendChartData ( $statistics->getOsGraph ( $_REQUEST[ 'begin_date' ] ,
                                            $_REQUEST[ 'end_date' ] ) );
}
?>
```

Quando a classe 'statistics.php' está a ser acedida externamente espera ser acedida através de uma variável 'graph' na URL. Esta variável pode tomar 4 valores, sendo eles respectivos aos 4 gráficos que retorna.

Para cada um dos gráficos é criada uma instanciação da classe, e procede-se à chamada das funções que retornam o array de dados correspondente. Com esse array, e utilizando a função 'SendChartData()' que faz parte da Classe Charts é mostrado o gráfico na página.

Termina assim o ficheiro 'statistics.php'.

Funções de datas

No decorrer da Classe 'statistics.php' certamente notou que são utilizadas funções que ainda não estão desenvolvidas. Essas funções vão transformar os formatos de data que utilizamos na Base de Dados.

Como na Base de Dados a data está a ser guardada no formato 'AAAA-MM-DD' de forma a tornar mais universal o Script, e estarmos mais habituados ao formato Português 'DD-MM-AAAA' criamos uma função que inverte a data pretendida para o formato Americano.

Cria-se então o ficheiro responsável pelo vários tratamentos necessários às datas utilizadas no script. O ficheiro a criar terá o nome de 'date_functions.php'.

```
<?php

function date_validation ( $date_to_check ) {
  $valid = true;

  if ( ( substr_count ( $date_to_check , "/" ) ) != 2 ) {
    $valid = false;
  }
```

Para verificar se uma data é válida irá proceder a uma série de testes, o primeiro irá verificar se a data introduzida tem apenas duas barras "/". Caso tenha mais ou menos que 2 barras, então estamos perante uma data inválida.

```
  else {
    if((ereg("([0-9]{1,2})/([0-9]{1,2})/([0-9]{4})",$date_to_check, $separated_date))){
      $day = $separated_date[ 1 ]; // atribuicao do dia
      $month = $separated_date[ 2 ]; // atribuicao do mes
      $year = $separated_date[ 3 ]; // atribuicao do ano
```

No 2º teste verifica-se se a data contém apenas números com o formato de 'DD-MM-AAAA'. Caso seja uma condição verdadeira, separa a data em 3 termos, sendo eles: o dia, o mês e o ano.

```
  if ( ( $year < 1900 ) || ( $year > 2200 ) ) {
    $valid = false;
  }
  elseif ( $month > 12 ) {
    $valid = false;
  }
```

Se o ano fôr inferior a 1900 ou superior a 2200, ou o mês fôr superior a 12 está perante uma data Inválida.

Seguidamente vai testar os dias possíveis de ter nos vários meses do ano.

```
  else {
    switch ( $month ) {
      case 2: // Fevereiro
        if ( $day > 29 ) {
          $valid = false;
        }
```

Se o dia fôr superior a 29 no mês de Fevereiro então terá uma data inválida.

```
        elseif ( ( $day == 29 ) && ( $year % 4 != 0 ) ) {
          $valid = false;
        }
        break;
```

Se tiver sido introduzido o dia 29, mas o ano não fôr bissexto então a data introduzida é inválida.

```
      case 4: // Abril
      case 6: // Junho
      case 9: // Setembro
      case 11: // Novembro
        if ( $day > 30 ) {
          $valid = false;
        }
        break;
```

Para todos os meses que têm 30 dias (Abril, Junho, Setembro, Novembro), se o dia introduzido fôr maior que 30 é então uma data inválida.

```
      case 1: // Janeiro
      case 3: // Marco
      case 5: // Maio
      case 7: // Julho
      case 8: // Agosto
      case 10: // Outubro
      case 12: // Dezembro
        if ( $day > 31 ) {
          $valid = false;
        }
        break;
      }
```

Se o dia fôr superior a 31 nos restantes meses (Janeiro, Março, Maio, Julho, Agosto, Outubro, Dezembro) é também uma data inválida.

```
    }
   }
  }
  return $valid;
}
```

Se em qualquer altura dos testes se verificar que a data é inválida o estado da variável '$valid' passa a 'false' indicando assim que a data é inválida. Se a data passar por todos os testes com condição positiva então o retorno da variável será 'true'.

```
function changeDate ( $inputDate ) {
        $split = explode( "/", $inputDate );
        $outputDate = "" . $split[2] . "/" . $split[1] . "/" . $split[0];
        return $outputDate;
}
```

Com a função 'changeDate()' procede-se então à alteração do formato da data tal como já foi referido anteriormente.

```
function get_dates ( ) {
  $end_date=date("d/m/Y");
  $begin_date=date("d/m/Y",strtotime("-1 week",strtotime(changeDate($end_date))));

  $date_interval[ 'begin_date' ] = $begin_date;
  $date_interval[ 'end_date' ] = $end_date;

  return $date_interval;
}
```

Na página principal do script existe a possibilidade do utilizador seleccionar as datas que pretende. No entanto, por defeito essas datas estão preenchidas com uma semana anterior ao dia actual.
Nesta função verifica-se qual é o dia actual e guarda-se essa data, verifica-se o qual o dia há uma semana atrás dessa data e retornam-se as duas.

```
function date_interval_form ( $begin_date , $end_date ) {

  $form = "<form action='".$_SERVER['PHP_SELF']."' method='post' target='_self'
                                                    id='frmIntervaloDatas'>
            <div align='center'>
              <br /><table width='600' border='0' cellspacing='0' cellpadding='0'>
           <tr class='big_titles'>
             <td height='40' colspan='5' align='center'>
               Per&iacute;odo a que correspondem as estatisticas
             </td>
           </tr>
           <tr>
             <td width='100' height='40' align='center' class='item'>
               Data In&iacute;cio:
             </td>
             <td width='150'>
               <input name='begin_date' type='text' id='begin_date'
                                    maxlength='10' value='".$begin_date."' />
             </td>
             <td width='100' class='messagem' align='center'>
               Formato:<br />(dd/mm/aaaa)
             </td>
             <td width='100' class='item' align='center'>
               Data Fim:
             </td>
             <td width='150'>
               <input name='end_date' type='text' id='end_date'
                                   maxlength='10' value='".$end_date."' />
             </td>
           </tr><tr>
             <td height='40' colspan='5' align='center'>
               <input name='btnOK' type='submit' id='btnOK' value='Gerar Estatisticas'/>
             </td>
           </tr></table></div></form>";

      return $form;
}
?>
```

De forma a tornar mais simples a integração do script em qualquer página, cria-se uma função que permite criar o formulário de selecção do intervalo de datas.

Termina então a criação do ficheiro 'date_functions.php'.

Ligação à Base de Dados

Para aceder à informação relativa à base de dados de forma rápida e eficiente cria-se um ficheiro 'connection.php' que será guardado na raiz do website. Este ficheiro terá todas os dados relevantes à ligação da Base de Dados utilizada para o bom funcionamento do script.

```
<?php

  $hostname = "localhost";
  $database = "nome_da_bd";
  $username = "utilizador";
  $password = "password";
  $connection = mysql_pconnect($hostname, $username, $password) or
                trigger_error(mysql_error(),E_USER_ERROR);

?>
```

Página principal

Agora que está construído o script, falta apenas pô-lo a trabalhar.

```
<?php

  include ( "statistics.php" );
  include ( "date_functions.php" );

  $statistics = new Statistics;
  $statistics -> saveStatistics ( );
?>
```

Este será o código que deverá ser colocado em todas as páginas que queira incluir no sistema de estatísticas. Apenas este código será responsável pela recolha de dados do visitante, guardando-os na base de dados para posterior tratamento.

Agora para realizar a página de resumo das estatísticas:

```
<?php

if ((isset($_POST['begin_date'])) && (isset($_POST['end_date']))){
  if(date_validation($_POST['begin_date']) && date_validation($_POST['end_date'])){
    if ( $_POST[ 'begin_date' ] > $_POST[ 'end_date' ] ) {
      $date_interval = get_dates ( );
      $begin_date = $date_interval[ 'begin_date' ];
      $end_date = $date_interval[ 'end_date' ];
    } else {
      $begin_date = $_POST[ 'begin_date' ];
      $end_date = $_POST[ 'end_date' ];
    }
  }
} else {
```

```
  $date_interval = get_dates ( );
  $begin_date = $date_interval[ 'begin_date' ];
  $end_date = $date_interval[ 'end_date' ];
}
?>
```

Coloca-se este bloco de código no topo da página que queira ver incluída no sistema, antes do código HTML. Este conjunto de código será responsável por testar se o utilizador já introduziu datas, ou se deverá ser utilizado as datas por defeito.

As datas que o utilizador eventualmente tenha introduzido serão validadas e apenas serão utilizadas caso se verifiquem válidas, de outra forma serão utilizadas as datas por defeito.

De seguida são demonstradas as instruções para completar a página com:

Menu de introdução de datas

```
<?php echo date_interval_form ( $begin_date , $end_date ); ?>
```

Gráfico de Total de visitas por dia

```
<?php echo InsertChart ( "charts.swf", "charts_library",
"statistics.php?graph=1&begin_date=".$begin_date."&end_date=".$end_date ,390 ,250
,"F8F8F8", false); ?>
```

Resumo de Páginas visitadas

```
<?php echo $statistics -> getPageVisits ( $begin_date , $end_date ); ?>
```

Conclusão

Termina-se desta forma este tutorial passo-a-passo que permite de uma forma simples, conhecer as informações essenciais de um visitante tais como, País, de onde vêm entre outras.

Existem uma 'infinidade' de sistemas on-line que permitem controlar as visitas a um website de forma eficaz, como por exemplo, o Google Analytics, mas o defeito destes sistemas, é a quantidade excessiva de informação desnecessária para uma visualização rápida.

A enorme vantagem deste sistema é a possibilidade de ser totalmente integrado num website, que se traduz numa mais valia, pois o utilizador poderá optar por partilhar a informação com todos os visitantes do seu website ou manter esta informação numa área de acesso reservado.

Quem quiser fazer download do script completo pode aceder a http://www.nulldesign.online.pt/project/estatisticas/ .

JNI Java Native Interface

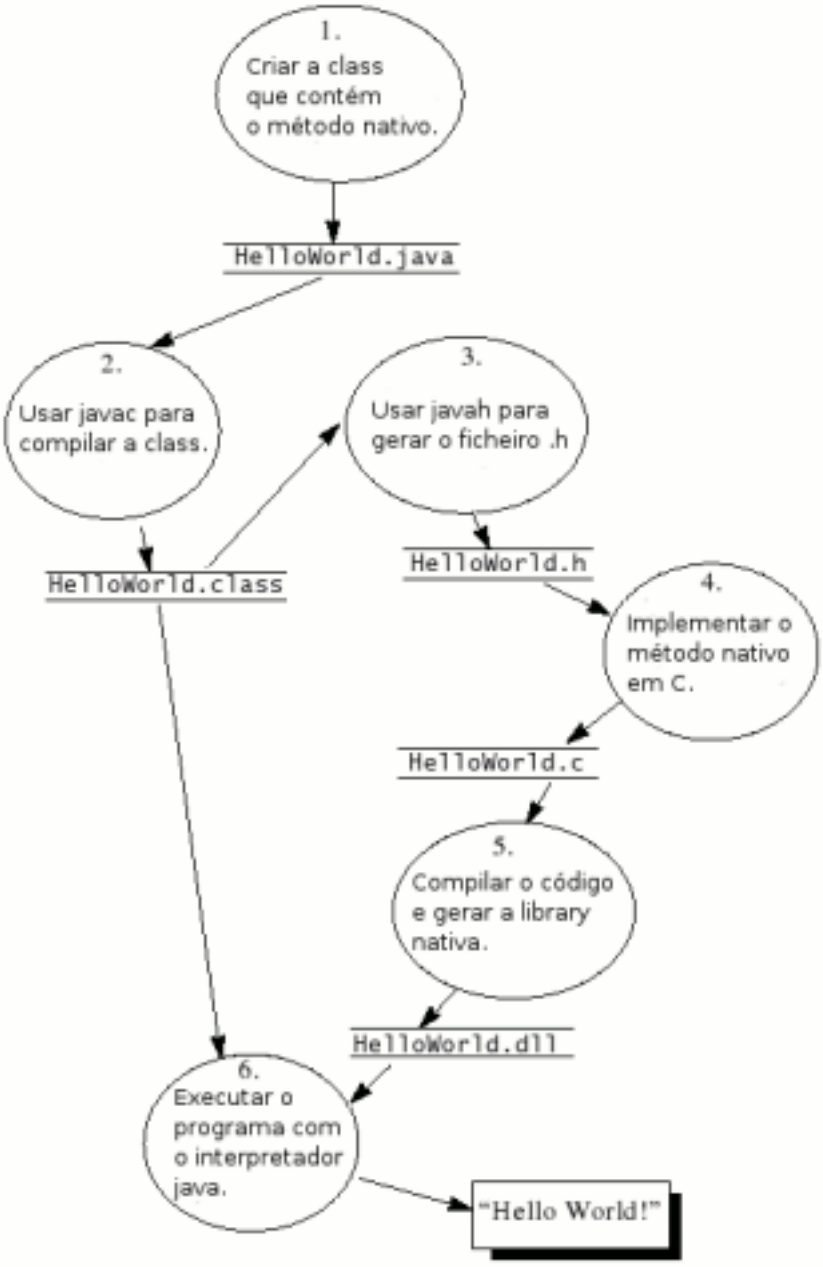
Java Native Interface (JNI) é um framework de programação que permite o código executado pela Java Virtual Machine chamar e ser chamado por aplicações nativas (aplicações específicas para um hardware e sistema operativo) e por bibliotecas escritas em outras linguagens, tais como C, C++, Delphi e Assembly.

Como todos os frameworks JNI também tem as suas implicações, o uso do JNI coloca em risco dois benefícios da plataforma Java. Primeiro as aplicações Java que utilizam JNI ficam dependentes do ambiente de execução, contrário de Java puro que pode ser executado em múltiplos ambientes com JNI a parte nativa tem de ser recompilada em cada um dos ambientes de execução, perdendo a portabilidade do Java.

Segundo a linguagem de programação Java é uma linguagem type-safe e segura, ao contrario das linguagens nativas como C ou C++ que não são seguras.

Por estas razões o uso de JNI deve ser feito de forma cuidada e apenas caso não seja possível implementar apenas com a linguagem Java.

Depois desta breve introdução vamos passar então à prática, vamos ver quais os passos necessários para a criação de um programa em Java com JNI.



Depois de vermos o percurso a percorrer vamos começar com a implementação da classe HelloWorld.java

```
public class HelloWorld{
  private native void print();
  public static void main(String
args[]){
    new HelloWorld().print();
  }
  static{
    System.loadLibrary("HelloWorld");
  }
}
```

Na linha 3 foi declarado o método nativo print e nas linhas 9 e 10 é dito ao Java para carregar o biblioteca "HelloWorld", que vamos criar depois de realizar mais alguns passos. Em seguida vamos compilar a classe criada : javac HelloWorld.java.

Depois de obtermos o ficheiro HelloWorld.class vamos usar a ferramenta javah do Java para criar o ficheiros header (HelloWorld.h), com o comando javah -jni HelloWorld vamos obter o ficheiro HelloWorld.h com o seguinte conteúdo.

```
/* DO NOT EDIT THIS FILE - it is
machine generated */
#include <jni.h>
/* Header for class HelloWorld */

#ifndef _Included_HelloWorld
#define _Included_HelloWorld
#ifdef __cplusplus
  extern "C" {
#endif
/*
 * Class:     HelloWorld
 * Method:    print
 * Signature: ()V
 */

JNIEXPORT void JNICALL
Java_HelloWorld_print (JNIEnv *,
jobject);

#ifdef __cplusplus
  }
#endif
#endif
```

Como podemos ver na linha 15 é declarado o header para o método nativo que iremos implementar, vamos usar essa header para criar o ficheiro de código nativo (HelloWorld.c), como podemos ver em seguida.

```
#include <jni.h>
#include <stdio.h>
#include "HelloWorld.h"

JNIEXPORT void JNICALL
Java_HelloWorld_print(JNIEnv *env,
jobject obj){
  printf("Hello World!\n");
  return;
}
```

Se observarmos vamos ver que a linha 15 do ficheiro HelloWorld.h é praticamente igual ao código na linha 5 do ficheiro HelloWorld.c.

Agora que já dispomos de todos os ficheiros necessários "HelloWorld.java", "HelloWorld.class", "HelloWorld.h" e "HelloWorld.c" vamos criar a nossa biblioteca nativa. Vamos ver como o fazer em diferentes sistemas operativos.

Linux com GCC

```
gcc -fPIC -I jdk/include -I
jdk/include/linux -shared -o
libHelloNative.so HelloNative.c
```

Solaris com Sun Compiler

```
cc -G -I jdk/include -I jdk/include/solaris -o
libHelloNative.so HelloNative.c
```

Windows com MS C++ Compiler

```
cl -I jdk\include -I
jdk\include\win32 -LD HelloNative.c -
FeHelloNative.dll
```

Onde jdk é o caminho para a pasta de instalação do Java.

Vamos agora ver como realizar a passagem de dados entre Java e C através do JNI. Para isso criamos um pequeno programa em que vamos imprimir uma pergunta, usar a função scanf do C para recolher o input e retorná-lo para o Java como um Objecto String. Para isso vamos começar claro pela criação do ficheiro Prompt.java.

```
public class Prompt{
  private native String getLine(String
prompt);

  public static void main(String args[]){
    Prompt p = new Prompt();
    String input = p.getLine("Qual
                      o seu nome ? ");
    System.out.println("Escreveu : "
                        +input);
  }

  static{
    System.loadLibrary("Prompt");
  }
}
```

Em seguida vamos gerar o ficheiro Prompt.h que irá conter algo semelhante a este:

```
/* DO NOT EDIT THIS FILE - it is
machine generated */
#include <jni.h>
/* Header for class Prompt */
#ifndef _Included_Prompt
  #define _Included_Prompt
  #ifdef __cplusplus
    extern "C" {
  #endif
  /*
   * Class:     Prompt
   * Method:    getLine
   * Signature: (Ljava/lang/String;)
   *             Ljava/lang/String;
   */
  JNIEXPORT jstring JNICALL
          Java_Prompt_getLine(JNIEnv *,
          jobject, jstring);

  #ifdef __cplusplus
    }
  #endif
#endif
```

Até aqui não há nada de novo relativamente ao HelloWorld implementado anteriormente, as novidades então agora na implementação do ficheiro Prompt.c que vai conter algumas novidade ao nível de JNI e no tratamento de Strings em JNI-C. Vamos então implementar o seguinte código.

```
#include <stdio.h>
#include <jni.h>
#include "Prompt.h"

JNIEXPORT jstring JNICALL
Java_Prompt_getLine(JNIEnv *env,
jobject obj, jstring prompt){

  char buf[128];
  const jbyte *str;

  str = (*env)->GetStringUTFChars(
                    env,prompt,NULL);

  if(str == NULL){
    return NULL;
  }

  printf("%s",str);
  (*env)->ReleaseStringUTFChars(
                     env,prompt,str);
  scanf("%s",buf);
  return (*env)->NewStringUTF(env,buf);
}
```

Como podemos ver na linha 5 nos argumentos do método para além das variáveis de ambiente que por defeito são colocadas, neste caso foi adicionada mais uma a variável prompt do tipo jstring que corresponde à String dada como argumento da classe Java. Como o C não tem o Objecto String como o Java vamos ter de passar a String para outra forma trabalhável em JNI-C, para isso é criada uma variável apontador do tipo jbyte, que de uma forma básica é o byte do JNI da mesma forma que o jstring é a String do JNI. O passo final para fazer a conversão é usar a função do JNI GetStringUTFChar que retorna uma cópia da String dada no segundo argumento (linha 10) e assim a variável str vai ficar com o valor da String dado como argumento ao método getLine.

Depois deste procedimento basta utilizar a função printf do C para imprimir a String e para terminar o processamento da String vamos usar a função ReleaseStringUTFChars (linha 17) que vai libertar a memória usada para guardar a String, atenção que este método apenas deve ser utilizado depois de terem terminado todas as operações na String.

Vamos agora para finalizar o nosso programa capturar o input do utilizador através da função scanf (linha 18) e guardar o input sob a forma de String no array de char definido na linha 7, em seguida basta converter o array de char para um Objecto String e retorná-lo para o Java, para isso vamos usar a função NewStringUTF como podemos ver na linha 19. Ao executar este exemplo o que vai acontecer será um intercâmbio de dados bastante simples, o Java envia uma String que é impressa pelo C, em seguida a função scanf vai capturar o input do utilizador e irá retornar esse input para o Java que o irá imprimir com o método println().

Passamos agora ao tratamento de arrays de Java para C, para isso vamos implementar o seguinte código.

```
public class IntArray{

  private native int sumArray(int []
arr);

  public static void main(String args
[]){
    IntArray p = new IntArray();
    int arr[] = new int[10];
    for(int i = 0; i < 10; i++){
      arr[i] = i;
    }
    int sum = p.sumArray(arr);
    System.out.println("sum = "+ sum);
  }

  static{
    System.loadLibrary("IntArray");
  }
}
```

Como podemos ver o código implementado vai criar um array com 10 posições e vai preenche las com os números de 0 a 9, em seguida vai executar o método nativo sumArray que vai somar todos os valores presentes no array e vai retornar o total da soma.

Vamos agora passar à implementação do método nativo e para isso vamos começar como sempre por gerar o ficheiro IntArray.h.

```
/* DO NOT EDIT THIS FILE - it is
machine generated */
#include <jni.h>
/* Header for class IntArray */

#ifndef _Included_IntArray
  #define _Included_IntArray
  #ifdef __cplusplus
    extern "C" {
  #endif

  /*
   * Class:     IntArray
   * Method:    sumArray
   * Signature: ([I)I
   */

  JNIEXPORT jint JNICALL
Java_IntArray_sumArray(JNIEnv *,
jobject, jintArray);

  #ifdef __cplusplus
    }
  #endif
#endif
```

Ao vermos os argumentos do método podemos verificar que está agora presente um jintArray que corresponde ao array de inteiros do Java que é dado como argumento na chamada do método.

Agora o código nativo do nosso método.

```
#include <jni.h>
#include <stdio.h>
#include "IntArray.h"

JNIEXPORT jint JNICALL
Java_IntArray_sumArray(JNIEnv *env,
jobject obj, jintArray arr){

  jint *carr;
  jint i = 0;
  jint sum = 0;
  carr = (*env)->GetIntArrayElements(
                        env,arr,NULL);
  if(carr == NULL){
    return 0;
  }

  for(i = 0; i < 10; i++){
    sum += carr[i];
  }

  (*env)->ReleaseIntArrayElements(
                       env,arr,carr,0);

  return sum;
}
```

Analisando o código podemos ver que na linha 7 é criado um apontador do tipo jint que vai ser usado para guardar o array dado como argumento ao método, para isso usamos a função Get<Tipo>ArrayElements em que o <Tipo> é substituído neste caso por Int, esta função vai retornar um apontador para o array dado como argumento como se pode ver na linha 10.

Nas linhas 15 a 17 é executado um ciclo for(), que faz a soma de todo o array e guarda o valor na variável sum previamente criada. Depois disto basta usar a função Release<Tipo>ArrayElements para libertar o apontador que contém o array (linha 19) e por fim na linha 21 vamos retornar a soma.

O JNI tem entre outras uma função bastante importante para o tratamento de arrays, trata-se da função GetArrayLength que como o próprio nome indica retorna o tamanho de um array, esta função pode ser usada da seguinte forma :

```
jsize len = (*env)->GetArrayLength(
                                env, arr);
```

Conclusão

E com isto encerramos este artigo sobre Java Native Interface, como podem ver é um framework muito poderoso mas mas que deve ser apenas usado como recurso à falta de outro recurso por parte do Java ou no caso de criação de novas bibliotecas. Em todo o caso deve ser utilizado com bastante cuidado visto tratar-se de programação não segura.

Fábio Correia

LUA Linguagem de Programação

Provavelmente se o leitor ouvir alguém falar em LUA, irá logo pensar no planeta satélite que o astronauta Neil Armstrong teve o privilégio de pisar pela primeira vez. No entanto neste artigo iremos falar de algo bem diferente, mais ao alcance de uma pessoa comum…

História e Caracaterísticas

A LUA foi desenvolvida em língua Portuguesa por uma equipa de programadores do departamento Tecgraf da Universidade Católica do Rio de Janeiro, em código open-source. O objectivo inicial foi a gestão de um projecto industrial da Petrobras.

Graças à sua portabilidade, facilidade de aprendizagem, rapidez e eficiência, rapidamente cresceu para outras áreas de programação, estando actualmente a fazer parte de aplicações informáticas de grande porte.

A LUA é uma linguagem de programação procedural do género de Pascal, C, Python, PHP, Cobol, entre outras. Por este facto qualquer programador terá uma curva de aprendizagem curta e rápida. No entanto a maior vantagem da LUA é talvez a sua enorme portabilidade que lhe permite interacção com qualquer outra linguagem de programação ou sistema operativo.

Com a sua sintaxe simples, qualquer script é de desenvolvimento rápido, e com um menor tempo de processamento, talvez tenha sido esta característica que cativou a grande Empresa de Jogos para PC "Lucas Arts".

Quem não se lembra do grande título de jogo "Escape from Monkey Island" na década de 90. Provavelmente o que o leitor não sabe é que LUA foi utilizada nesse jogo para o desenvolvimento da programação responsável pelo controlo da personagem principal, todos os cálculos necessários à movimentação das personagens eram realizados através de scripts em LUA.

Com a entrada no mundo dos jogos, a rapidez e portabilidade da linguagem despertou a atenção de outras Entidades. Os Estados Unidos da América foi a próxima conquista, tendo inúmeras Empresas do Parque Industrial de Silicon Valley optado pela LUA para desenvolver grande parte do seu software.

Posteriormente a afirmação final de LUA é consolidada com a interacção dos sistemas do programa espacial da NASA e com alguns dos projectos internos .NET da Microsoft.

Mais informação em:
Página oficial da Linguagem LUA
http://www.lua.org
Tecgraf
http://www.tecgraf.puc-rio.br/

A evolução de LUA

Com um crescimento bastante acentuado, cada vez mais programadores tiveram a curiosidade de ler e se informar o porquê de tanto se ouvir falar em LUA. Rapidamente os grupos de debate da comunidade foram aumentando e os projectos novos surgiam a uma rapidez alucinante. Por tudo isto a evolução de LUA teria de passar por algo que se tornou indispensável no nosso dia-a-dia…. a Internet. Capaz de desenvolver Websites completos através da biblioteca CGI/LUA e LuaSQL.
Imagine o que será desenvolver um website completamente em CGI/LUA (equivalente para a LUA, tal como o IIS é para o ASP) com recurso a LuaSQL.

Com o Back-end compilado em ANSI C ou Lua pré-compilado, se um dia decidir trocar o servidor de Windows para Solaris por exemplo, não será necessário desenvolver a plataforma toda de novo, basta recompilar o código C e configurar o novo Webserver.

Mais informação em:
Módulo Lua para o Apache
http://oss.digirati.com.br/mod_lua/1.0/
Plataforma de Desenvolvimento de aplicações WEB em LUA ht
tp://www.keplerproject.org/
Alojamento de projectos em LUA ht
tp://luaforge.net/
Comunidade de utilizadores LUAhtt
p://lua-users.org/

A programação

Como este artigo não pretende demonstrar a totalidade da linguagem de programação LUA, mas sim abordar as principais características que tornam esta linguagem única, recomenda-se a leitura do manual em http://www.lua.org .

Tal como existe no JAVA a Java Virtual Machine, também LUA tem algo parecido, que tem o nome de Ambiente Global Único, este ambiente guarda todas as variáveis globais e definições de funções. É automaticamente activado quando o interpretador é inicializado, e está activo durante toda a execução do script. O Ambiente Global Único é manipulado apenas por código escrito em LUA ou por bibliotecas em C, que utilizem funções da interface C-Lua.

Variáveis

Tal como acontece em PHP ou Python as variáveis globais não precisam de ser declaradas, basta utilizar algo do género:

```
a = 2.3;
```

É importante referir que as variáveis globais não têm tipos de dados definidos, apenas o conteúdo define o tipo de variável, existindo sete tipos básicos: nil, number, string, function, cfunction, userdata e table.

Atribuição múltipla de variáveis

De forma a maximizar o código, com a LUA podemos fazer atribuições múltiplas de variáveis:

```
n, i = "Olá, Mundo" , 2;
```

Ficando desta forma a variável n com a string "Olá Mundo" e a variável i com o valor "2".
Troca de valores entre atribuições múltiplas
É possível no entanto a atribuição múltipla com troca de valores entre variáveis:

```
a, b = b, a;
```

Fazendo com que a receba o valor anteriormente armazenado por b e que b, receba o valor anteriormente armazenada por a, sem necessidade de variáveis temporárias. Imagine o que será os velhos algoritmos de bubblesort para ordenação de dados com esta vantagem.

Concatenação

Para se poder concatenar variáveis com conteúdos string utiliza-se o operador "..":

```
a = "Olá";
b = ", Mundo";
z = a .. b;
```

Resultado em z será "Olá, Mundo".

Variáveis Locais

Quando se executa ciclos (if..else..end, repeat..until, do..while, etc..) existe a possibilidade de criarmos variáveis locais disponíveis apenas dentro desse ciclo.

```
a = 2;
if a > 0 then
   local a = ( a + 2 ) * 5;
   print( a );
end
print( a );
```

O resultado do primeiro print é 20 e do segundo print é 2. Com esta característica de LUA, podemos criar uma variável que apenas dura enquanto o ciclo existir, podendo realizar as operações que quisermos mesmo que a variável tenha o nome igual, e quando sairmos do ciclo a variável original continua com o valor inicial.

Múltiplo retorno em funções

Da mesma forma que é possível realizar múltiplas atribuições em variáveis podemos ter múltiplos retornos de uma função e aliar a múltiplas variáveis:

```
function num_anterior_e_seguinte ( x )
   anterior = x -1;
   seguinte = x +1;
   return anterior , seguinte;
end

num_actual = 5;
anterior, seguinte =
num_anterior_e_seguinte( num_actual );
```

Com este código podemos fazer uso de uma função que nos irá retornar os números que antecede e segue o introduzido na função. O resultado será: a variável anterior irá conter o número 4 e a variável seguinte irá conter o número 6.

Recursividade

Em linguagens procedurais a recursividade é por vezes muito utilizada, o seu uso em LUA é algo simples e rápido o que torna a sua utilização muito grande:

```
function factorial( n )
   if n == 0 then
      return 1;
   else
      return n * factorial( n -1 )
   end
end
```

Com a utilização desta função pode-se verificar que a utilização da recursividade é algo simples de implementar. Para programadores de Java esta característica não é inovadora, mas em grande parte de linguagens de programação, a recursividade de funções é algo que ainda não existe, obrigando os programadores a um esforço extra no código para realizar o mesmo objectivo.

Mais informação em:
Manual da Linguagem LUA 5.0 (ENG)
http://www.lua.org/ftp/refman-5.0.pdf
Manual da Linguagem LUA 3.1 (PT)
http://www.lua.org/ftp/nocoes-3.1.pdf

Instalação e ambiente de desenvolvimento

Nesta fase o leitor certamente deve estar desejoso de experimentar a LUA. Então vamos primeiro proceder à instalação. Para poder instalar o Ambiente Geral Único de LUA deverá realizar o download do Interpretador em:
http://prdownloads.sourceforge.net/luacheia/luacheia5-win32-5.0.1a5.exe?download

Existem várias opções e versões de download, no entanto vamo-nos centrar na instalação para Windows. Após realizar o download do interpretador vamos então instalar a LUA mantendo o directório sugerido por defeito (c:\luacheia\). Após alguns cliques mais, e estamos aptos a programar em LUA. Para desenvolver código em LUA qualquer editor de texto serve o propósito, até mesmo o simples bloco de notas (Notepad) do Windows, no entanto sugiro a utilização do famoso IDE Eclipse e a instalação do módulo LuaEclipse em: http://luaeclipse.luaforge.net/br/index.html.

O primeiro script de LUA

Para fazermos cumprir a tradição o nosso primeiro script será o famoso "Olá Mundo!" aparecer no nosso ecrã. Para tal abrimos o Bloco de notas do Windows e incluímos o seguinte código:

```
print( "Olá Mundo!" );
```

E está concluído o nosso script. Simples, não é? Ao contrário de várias linguagens que nos obrigam a seguir uma estrutura, em LUA tudo é em função da rapidez…seja ela de processamento ou desenvolvimento.

No fim de guardar na pasta C:\luacheia\ e de aceder pelo DOS (escrevendo cmd no menu executar) basta escrever cd \luacheia\ seguido de luacheia5 olamundo.lua. E aparece-nos no ecrã Olá Mundo!.

O utilizador agora apenas tem de dar largas à sua imaginação e começar a desenvolver em LUA.

Bruno Vaz

CSS: Cascading Style Sheet

Introdução

Hoje em dia, com a globalização das tecnologias e sobretudo da Internet, poucos são aqueles que nunca consultaram uma página web. Todavia, a sua grande maioria não imagina como é elaborada, e muito menos que por detrás daquela bela aparência existe, principalmente, uma linguagem de programação chamada HTML. Contudo, essa linguagem, uma vez que foi concebida para usos diferentes dos actuais, possui algumas limitações, e muita da "beleza" que ela proporciona muito se deve ao seu aliado CSS.

O que é?

O CSS é, então, a sigla inglesa de Cascading Style Sheet, que em português significa folha de estilo em cascata, e é nada mais do que um mecanismo para adicionar estilos (cores, espaçamentos, avanços, tipos de letra, backgrounds, etc) aos documentos web. Por outras palavras, podemos dizer que enquanto o HTML tem como função estruturar os textos, cabeçalhos, botões, tabelas, links, imagens, ou seja, o conteúdo da página, o CSS encarrega-se de definir as cores, o posicionamento, o estilo das linhas, isto é, tudo que diz respeito ao visual da mesma. Por sua vez, esta tarefa de estilização, independentemente da forma de definição (externa, incorporada ou inline) é sempre feita aparte do HTML, porção esta que contém todas as regras de estilo para os elementos do documento.

Neste sentido, este complemento do HTML torna a tarefa do programador bastante mais simplificada, visto que este pode alterar propriedades de várias páginas, alterando apenas um respectivo campo na folha de estilos.

Regras e Sintaxe

Uma folha de estilos não é mais do que um conjunto de regras CSS. Por seu turno, uma regra CSS é uma declaração com uma sintaxe própria constituída, na sua forma mais simples, por três partes: um seleccionador (elemento HTML identificado por uma tag, classe, id, etc, à qual será aplicada a regra em questão – exemplo: <body>, <h1>, .minhaclasse), uma propriedade (qualidade à qual vai ser aplicada a regra – exemplo: font, border, color – no elemento HTML) e um valor (característica adoptada pela propriedade – exemplo: letra tipo verdana, linha dupla, fundo branco), como se exemplifica de seguida:

```
seleccionador {
  propriedade: valor;
}
```

Como podemos verificar, na sintaxe de uma regra CSS primeiro escreve-se o seleccionador, e posteriormente a propriedade e o valor separados por dois pontos e entre chavetas. Contudo, se pretendermos definir mais de uma propriedade para um determinado elemento, é costume e hábito (embora seja opcional) utilizar-se o ponto e vírgula para as separar, e usar linhas distintas para as escrever, de forma a criar uma melhor legibilidade.

Neste sentido, de forma a elucidar o leitor, neste momento torna-se interessante o entendimento dos exemplos dados de seguida:

```
body {
  background: #999999;
}
```

No exemplo acima, o seleccionador é todo o documento (body), a propriedade é o fundo e o valor é a cor cinza.

```
h1 {
  font-size: 40px;
  font-style: italic;
  color:#00FF66;
  font-family: verdana;
  text-align: center;
}
```

Neste exemplo mais elaborado, o elemento é o "título1" (h1), e as cinco propriedades com os respectivos valores, fazem com que o resultado seja que todos os títulos principais (seguindo as características pela ordem em que são declaradas) tenham um tamanho de 40 pixeis, sejam em itálico, de cor verde, com o tipo de letra verdana e alinhados ao centro. O resultado pode ver-se imediatamente:

Contudo, nem só as tags (etiquetas) do HTML podem ser estilizadas. Com a evolução do CSS1 para o CSS2, para além de aumentarem o número de propriedades possíveis de personalizar, passou-se também a poder criar livremente uma classe e aplicá-la a qualquer elemento HTML. Para tal, basta que se omita o tag a personalizar, e que se coloque um ponto antes do nome da classe que pretendemos criar. Por exemplo:

```
.minhaclasse {
  color: #FF0000;
  text-align: right;
}
```

Assim, está criada a classe, pelo que poderá agora aplicá-la a qualquer elemento do documento web:

```
<h1 class="minhaclasse">
  Título vermelho e alinhado à direita.
</h1>
<p class="minhaclasse">
  Parágrafo vermelho e alinhado à
direita.
</p>
```

Resultado:

Titulo vermelho e alinhado à direita

Parágrafo vermelho e alinhado à direita

Seguindo a linha das classes, torna-se igualmente importante referir que uma mesma tag pode ser personalizada com propriedades diferentes. Por outras palavras, as potencialidades do CSS permitem que no mesmo documento tenhamos, por exemplo, dois tipos de títulos principais (h1): um com letras vermelhas e outro com letras azuis. Para isso, basta que especifiquemos atrás do ponto da classe por nós criada a etiqueta HTML que pretendemos alterar. Vejamos o seguinte exemplo:

```
h1.vermelho {
  color: #FF0000;
}

h1.azul {
  color: #0000CC;
}
```

Na nossa folha de estilos definimos, então, dois estilos diferentes para o título principal, pelo que no exemplo abaixo se demonstrará como seriam aplicados na folha HTML:

```
<h1 class="vermelho">
  Título vermelho.
</h1>
<h1 class="azul">
  Título azul.
</h1>
```

Título vermelho.

Título azul.

Por fim, relativamente aos seleccionadores ID, estes diferem das classes na medida em que somente podem ser aplicados a um elemento HTML dentro do documento. Por outras palavras, os ID são classes também por nós criadas mas que se podem utilizar apenas num elemento. A sintaxe de um seleccionador ID é muito semelhante a uma classe, tendo apenas de novo o #, conforme podemos visualizar abaixo:

```
#meuID {
  propriedade: valor;
}
```

Tipos de vinculação de folhas de estilo

Adquiridas as regras e sintaxe desta "linguagem" de estilização, o leitor deve, neste momento, indagar-se como é que, afinal, o CSS e o HTML se associam, visto estes serem puros aliados. Na verdade, as folhas de estilo podem ser vinculadas aos documentos de três formas distintas:

1- Externa;
2- Incorporada;
3- Inline;

Folha de estilo externa:

Tal como o próprio nome indica, no método externo a personalização é feita num documento diferente do documento HTML, tendo esse arquivo (podendo ser elaborado em qualquer editor de texto) a extensão .css. Assim, depois do documento CSS criado, define-se na folha HTML um link que carregará as personalizações por nós elaboradas. O exemplo abaixo demonstra a sintaxe a colocar no documento para que se torne possível carregar a folha de estilos:

```
<head>
  ……..
  <link href="folhadeestilo.css"
rel="stylesheet" type="text/css" />
  ……..
</head>
```

Folha de estilo incorporada:

Uma folha de estilo diz-se incorporada quando a estilização é feita no próprio documento HTML. As regras, válidas somente para aquele documento, são declaradas no cabeçalho (head) com a tag <style>, como se pode observar de seguida:

```
<head>
  ……..
  <style type="text/css">
  <!--
    body {
      background: #999999;
    }
    p {
      font-size: 10px;
      color: #000000;
    }
  -->
  </style>
  ……..
</head>
```

Folha de estilo inline:

Uma folha de estilo é dita inline, quando a personalização é feita dentro da tag do elemento HTML. Contudo, este método é bastante limitado, pois perde a grande vantagem do poder do CSS, isto é, como só é aplicável aquele próprio elemento, não se pode com uma única instrução modificar vários elementos de vários documentos com o mínimo do esforço.
Contudo, a sua sintaxe pode ver-se a seguir:

```
<p style="color:#999999; font-size:20px;">
Isto é um parágrafo de cor cinza e com
um tamanho de letra de 20 pixeis.
</p>
```

Agora que já tem umas razoáveis noções de personalização CSS, o leitor pode começar por criar páginas simples HTML e personalizá-las a seu gosto.

Pode, também, aprofundar mais os seus conhecimentos em:
http://www.maujor.com/
http://www.codigofonte.com/css/

Sandro Pinto

Revista Linux

A Verdadeira Revista Portuguesa de Linux

Edição Bimestral em formato PDF

Download Gratuito

www.revista-linux.com

Segurança em Sistemas Distribuídos

Já lá vai o tempo em que era necessário comprar um supercomputador para realizar tarefas mais complexas. Hoje em dia tudo faz mais sentido quando a mesma tarefa é processada em paralelo por vários computadores, ou quando os mesmos periféricos são acedidos por vários terminais em simultâneo. Daí o conceito de Sistema Distribuído.

O que é um Sistema Distribuído – Definição

Um sistema distribuído é um conjunto de computadores independentes, interligados através de uma rede de computadores, que se apresenta ao utilizador como um sistema único e consistente. Por exemplo, uma tarefa qualquer a executar, se for feita por um conjunto de computadores a computação é distribuída por todos e não apenas por um, não sendo necessária sobrecarga de processamento.

Uma tarefa qualquer, se divisível em várias subtarefas pode ser realizada em paralelo.

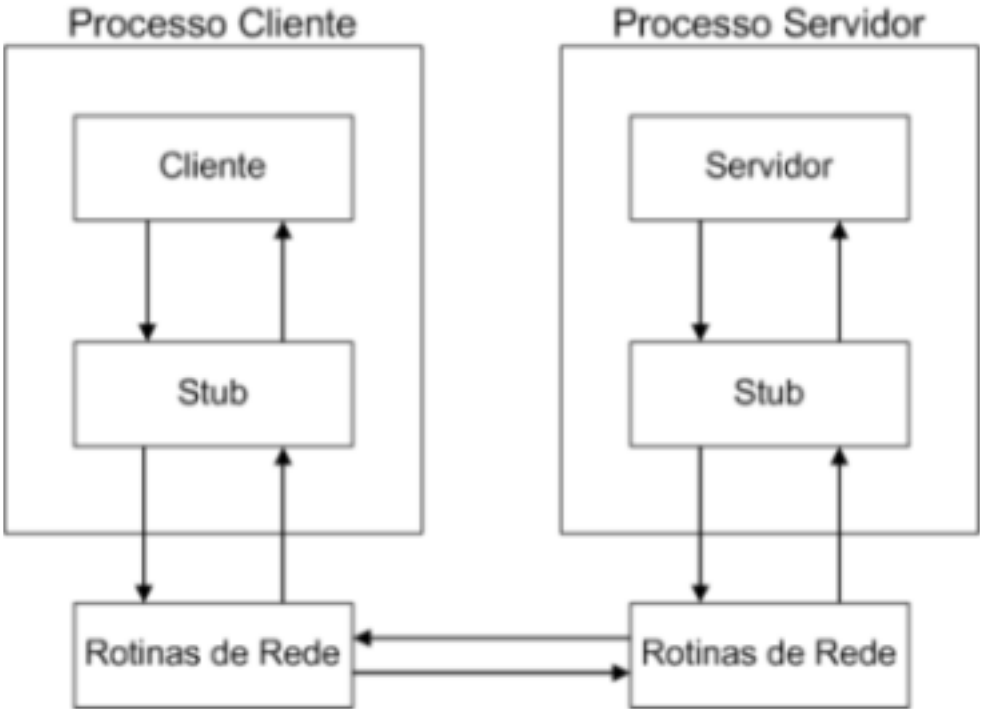
Vantagens / Desvantagens

A grande vantagem dos Sistemas Distribuídos é poder, com computadores baratos e de baixo processamento, formar um supercomputador que normalmente sairia muito caro.

O compartilhamento de recursos (impressoras, dados, CD´s, etc.), a transparência, concorrência (acesso das várias máquina a uma só), tolerância a falhas (confiabilidade - se falhar um computador numa rede de vários é mais fácil recuperar da falha do que se fosse apenas num computador), escalabilidade (facilmente podem ser adicionados computadores à rede), aumento da capacidade de computação, aumento da segurança / estabilidade global, factor económico (é melhor ter vários computadores medianos/fracos que um forte) e capacidade de processamento. São estas as principais vantagens.

Procedimentos Remotos

Os procedimentos remotos tiveram origem nos anos 80, surgindo da necessidade de haver uma metodologia de programação distribuída. A primeira foi o RPC (Remote Call Procedure), numa altura em que a linguagem dominante era o C. Como a linguagem C era procedimental e permitia chamar uma função em qualquer parte do código, pensou-se em desenvolver uma técnica que facilitasse o desenvolvimento de aplicações para Sistemas Distribuídos mantendo a ideia de chamar funções de máquinas exteriores da mesma maneira como se fosse na própria. Logicamente que tinha de haver algo que fosse totalmente transparente para o programador e que fizesse a conversão dos dados recebidos/enviados para o cliente e para o servidor, isto é, as stub routines. Para melhor perceber este procedimento, pode olhar para o seguinte esquema:



Como se pode observar, os stubs encarregam-se daquilo a que se chama marshalling, ou seja, converter os dados para o formato intermédio para enviar do cliente para o servidor ou do servidor para o cliente, e voltar a convertê-los do formato intermédio no local de destino (unmarshalling). Alguns exemplos de implementações de RPC são: Sun RPC (para plataformas UNUX / Linux), Java RMI (RPC para Java), DCOM (RPC para Windows), SOAP (utilizado por serviços da web) e CORBA (padrão RPC independente da plataforma).

Política de Segurança

Uma rede de computadores, por natureza, tem várias ameaças. Num sistema informático o principal objectivo é proteger a informação, ou seja, controlar a capacidade de aceder, alterar ou executar um ficheiro, um registo, uma base de dados, etc. Mas o principal objectivo de um sistema distribuído é a partilha de informação, logo isto conduz a um dilema porque também é necessário facilitar o acesso à mesma. A solução passa por definir uma política de segurança, na qual se deve ter sempre em conta vários factores como é o caso do custo, valor da informação, o que se pretende proteger, que mecanismos de protecção podem ser utilizados e claro, fazer uma avaliação das possíveis ameaças e formas de ataque. Esta política teve origem nos sistemas militares, em que o principal problema era a divulgação de informação não autorizada. Para além do controlo do acesso à informação também é necessário garantir a sua integridade. É necessário ter em consideração as várias ameaças a definir na política de segurança. Entre elas destacam-se o acesso e divulgação não autorizado de informação, modificação ilegítima da informação, operações não autorizados sob recursos do sistema, bloqueio de acesso legítimo à informação e vandalismo.

Quanto aos possíveis ataques, destacam-se: a personificação (quando alguém se faz passar por nós), execução de operações que contornam os mecanismos de segurança, introdução de código malicioso ou prejudicial (exemplo: vírus) nos sistemas informáticos, criação ou utilização de canais para comunicações ilegítimas (fuga de informação), escuta de mensagens e adulteração de mensagens (modificação, inserção, repetição de mensagens antigas…).

Tipos de Políticas de Segurança

Destacam-se três tipos de políticas de segurança. A primeira é a chamada de isolamento dos agentes. Caracteriza-se por limitar a interacção do agente à sua máquina virtual (como se fosse uma sessão num sistema operativo). Implica a identificação de cada agente, isto é, necessita de autenticar-se (por exemplo a validação recorrendo ao par login/password).

Cada agente possui um ID, isto é, um identificador que o representa cada vez que tentar aceder a um recurso do sistema. Esta política tem várias vantagens: implica a impossibilidade de contornar os mecanismos de controlo de segurança e de criar ou utilizar canais de comunicação encobertos ou ilegítimos. Também possui diversas desvantagens, algumas delas já mencionadas: infiltração de código malicioso, escuta de terminais de monitores (através da radiação electromagnética dos terminais é possível ver o que uma pessoa faz no seu posto de trabalho), e o ataque às palavras passes dos utilizadores (em que a maioria são baseadas em dicionários: séries de televisão, filmes, próprio nome, família, matrícula do carro, …).

Outro tipo de política é o chamado controlo de acessos e foca-se na limitação da capacidade de interacção. Existe uma matriz de direitos de acesso que dá autorização ou não a um agente quando este pretende efectuar uma operação. Também existem as ACLs (Access Control Lists), que em português corrente significa lista de controlo de acessos. Estas são armazenadas junto de cada objecto e controlam o acesso de agentes a objectos do sistema. Há também as capacidades de cada agente perante os objectos do sistema, mas estas capacidades são guardadas junto de cada agente. Depois é importante realçar a importância de dois mecanismos: autenticação (validação da identidade, uma falha na autenticação invalida a utilização da matriz de direitos de acesso) e autorização (operação que valida os direitos do agente sobre o objecto). Depois há uma espécie de Monitor de controlo de referência, que valida na altura de execução de uma operação se o agente tem o direito de a efectuar. Isto também obriga a que cada objecto seja alvo de uma identificação única e inequívoca. Os ataques a este tipo de política visam alterar o isolamento entre agentes, alterar a matriz de controlo de referência e eliminar o controlo efectuado pelo Monitor de Controlo e Referência.

O outro tipo de política é o controlo do nível de segurança da informação e parte do princípio que só se deve dar ao agente a informação que ele realmente precisa. A informação é classificada de acordo com o seu nível de confidencialidade. O controlo de acesso dos agentes é efectuado com base em duas regras: não podem ler informação classificada em níveis superiores ao seu, e não pode escrever informação classificada em níveis inferiores ao seu.

Criptografia

De forma de evitar a escuta e falsificação de mensagens surge a cifra de mensagens – Criptografia. Ou seja, é estabelecido um canal seguro onde passam os dados cifrados. Esta técnica tem um senão, pois não evita a reutilização de mensagens.
Convém referir que apesar das mensagens estarem cifradas, existe uma pequena probabilidade de um atacante deduzir a chave, embora esta seja muito pequena. O tamanho da chave de cifra é fulcral, pois uma chave pequena é mais facilmente descoberta (através de "força bruta" por exemplo) que uma chave maior.

Existem dois tipos de criptografia: Criptografia Simétrica e Criptografia Assimétrica. Os algoritmos de cifra simétrica são normalmente mais rápidos que os de cifra assimétrica, a chave só deve ser conhecida pelos intervenientes legítimos e é usada tanto nas operações de cifra como nas de decifra. Um exemplo disso é o Algoritmo DES (Data Encription Standard).
Na criptografia assimétrica usam-se pares de chaves relacionadas: uma chave pública (do conhecimento de todos) e outra privada (do conhecimento de uma entidade). Desta forma elimina-se o problema da partilha de chaves.

Para obter a privacidade dos dados cifra-se com a pública e decifra-se com a privada, para a autenticidade dos dados cifra-se com a privada e decifra-se com a pública. Um exemplo de um algoritmo que utiliza a criptografia assimétrica é o caso do Diffie-Helman, muito utilizado em cifra de comunicações em redes.

Autenticação, Autorização e Assinatura digital

Para a autorização de operações dos agentes em Sistemas Distribuídos, é usado o mecanismo mais vulgar em sistemas multiprogramados, como é o caso do UNIX / Linux. Ou seja, cada grupo e cada utilizador tem um código que é atribuído quando o utilizador se autentica. Associados a esse código existem os direitos positivos ou negativos às categorias de dono, grupo e restantes utilizadores.

Um factor de extrema importância é saber a autoria de determinada informação. A isto dá-se o nome de assinatura digital e processa-se de forma a calcular uma determinada Hash para a informação X associada ao Agente A. Com isto pretende-se garantir que o remetente da informação foi XPTO (autenticidade) e que o conteúdo não foi adulterado (garantir a integridade).

Conclusão

Com este artigo pretendeu-se que o leitor ficasse a saber, por alto, o que é um Sistema Distribuído e quais as suas características, nomeadamente: as ameaças, vantagens, desvantagens, algumas técnicas usadas na comunicação de processos, entre outras. Convém referir que nenhum tema foi aprofundado pelo que se quiser saber mais sobre algum assunto terá que pesquisar na Internet sobre o mesmo. Para facilitar a tarefa, a seguir estão várias hiperligações para sites que tratam do assunto.

Referências

http://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o_distribu%C3%ADda
http://pt.wikipedia.org/wiki/Criptografia
http://www.infowester.com/criptografia.php
http://pt.wikipedia.org/wiki/Chamada_de_procedimento_remoto
http://www.faqs.org/rfcs/rfc1831.html

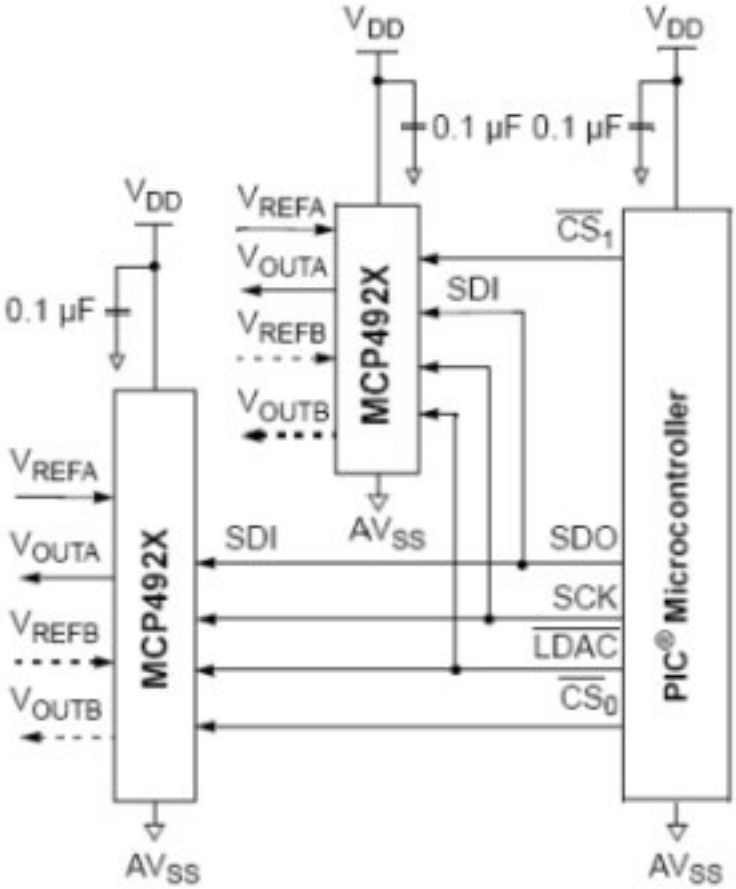
Pedro Teixeira

Microcontroladores: Protocolos SPI

Este artigo foi criado com o intuito de apresentar o protocolo SPI aplicado à comunicação entre microcontroladores e periféricos. Como interface será usada uma aplicação feita em Visual Basic.

Protocolos SPI

Visto os recursos disponíveis num microcontrolador serem limitados, por vezes há necessidade de os expandir. Assim existem circuitos integrados com as mais variadas funções: memórias EEPROM, shift registers, conversores A/D…
Para se poder comunicar com estes periféricos é necessário um protocolo de comunicação para que "ambas as partes se entendam". Dos muitos protocolos disponíveis, um que foi massivamente adoptado para comunicação entre microcontroladores e periféricos externos foi o SPI (Serial Peripheral Interface).



O protocolo SPI é um protocolo série que permite transmissão síncrona bidireccional de 8 bits.

O SPI necessita que exista um Master que controle o relógio. Visto o SPI ser bidireccional, o Master pode enviar ou receber informação dos slaves, mas este é que irá ditar quem envia ou quem recebe.

Para determinar o destinatário da informação é necessário uma ligação directa a cada um dos Slaves, que será a ligação que conecta os pinos CS (Chip Select). Assim quando a informação tiver como destinatário o Slave1, o pino CS1 terá que ser activado para que este aceite a informação.

Microcontroladores

Para demonstrar a aplicação deste protocolo será usado um Microcontrolador da Microchip (PIC18F2580) e uma DAC (MCP4921). Uma DAC é um dispositivo que converte um sinal digital para analógico (Digital-to-Analog Conversion), a DAC MCP4921 usa o protocolo SPI para receber a informação digital (8 bits) proveniente do microcontrolador e converte a informação recebida numa tensão analógica.

A DAC usada recebe a informação pelo pino SDI e embora o SPI permita que a comunicação seja bidireccional, não é possível enviar informação da DAC para o microcontrolador.

O comando de escrita da DAC consiste em 16 bits. Nestes 16 bits estará informação relativa ao modo de operação e aos dados. Os primeiros 4 bits dizem respeito ao modo de operação, os 12 bits restantes correspondem ao valor da tensão a converter.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
A/B	BUF	GA	SHDN	D11	D10	D9	D8

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
D11	D10	D9	D8	D7	D6	D5	D4

A fórmula de conversão da DAC é a seguinte

$$V_{OUT} = \frac{V_{REF} G D_N}{2^n}$$

A tensão de referência usada foi de 4.096V. Assim para uma tensão de 3,265V é necessário que os 12 bits menos significativos representem o número 3265, ou seja:

D11	D10	D9	D8	D7	D6
1	1	0	0	1	1

D5	D4	D3	D2	D1	D0
0	0	0	0	0	1

As funções que permitem que os registos sejam configurados são disponibilizadas pela Microchip, assim não será objectivo deste artigo explicar como as construir, mas sim como as usar, tal como aconteceu nos artigos relacionados com microcontroladores das edições anteriores.

```
#include <p18cxxx.h>
#include <math.h>
#include "usart_funcs.h"
#include "function.h"
#include <spi.h>
```

Para começar é necessário importar os protótipos das funções SPI

Antes de começar a usar os recursos do microcontrolador é necessário activar todos os módulos que serão usados. Assim activam-se as interrupções, as ADCs e define-se o tipo de saída da porta usada como Chip Select

```
void main(void)
{
  unsigned char dac_val1,dac_val2,
                          dac_val3;
  unsigned int node1,node3,node2,aux;


  //Enable interrupts
  INTCONbits.GIE = 1;
  INTCONbits.PEIE = 1;
  // Enable interrupt priority feature
  INTCONbits.IPEN = 1;
  // GIEH must be set in order to get
access to low priority interrupts
  INTCONbits.GIEH = 1;
  INTCONbits.GIEL = 1;


  adcInit();
  TRISAbits.TRISA4=0; //PORTA4->output
```

Activa-se de seguida a USART que será usada para comunicar com o PC e finalmente o módulo SPI.

```
//SYNC = 0, BRGH = 1, BRG16 = 0
//fosc=40Mhz,baudrate=115200-->spbrgh=21
  OpenUSART(USART_TX_INT_OFF
   & USART_RX_INT_OFF & USART_ASYNCH_MODE
   & USART_EIGHT_BIT & USART_CONT_RX
   & USART_BRGH_HIGH, 21);
  OpenSPI(SPI_FOSC_16, MODE_00,
SMPMID);
```

O programa irá aguarda por uma ordem vinda do PC, via USART. Quando for recebido algo pela USART o bit RCIF do registo PIR1bits ficará a "1".

Assim:

```
LATAbits.LATA4=1;
  while(1)
    {
      if(PIR1bits.RCIF)
      {
        switch (Re())
        {
          case 's':
```

Caso a informação recebida seja o carácter 's', então os próximos bytes a serem recebidos dizem respeito ao valor que se pretende converter.

```
dac_val1=Re();
dac_val2=Re();
dac_val3=Re();

dac_val1=convert_to_hex( dac_val1);
dac_val2=convert_to_hex( dac_val2);
dac_val3=convert_to_hex( dac_val3);

aux=0;
aux=dac_val2;
aux=aux<<4;
aux=aux|dac_val3;
```

Tendo em conta que os primeiros 4 bits dos 16 que serão enviados para a DAC dizem respeito ao modo de funcionamento, então faz-se uma máscara com o byte mais significativo para poder concatenar com os 4 bits de configuração.

```
LATAbits.LATA4 = 0;
WriteSPI(0b01110000 | dac_val1);
```

Os outros 2 bytes recebidos serão concatenados entre si para que se possam enviar para a DAC. De notar que antes de enviar qualquer informação a porta A4 foi colocada a zero para activar o CS da DAC. Embora possa ser contraditório, o CS da DAC é activado quando este estiver com o valor lógico "0"…

```
WriteSPI(aux);
LATAbits.LATA4 = 1;
break;
```

Se na vez de se ter recebido inicialmente o caracter "s", se se tiver recebido o carácter "r" então significa que se pretender ler o valor da ADC0, que faz parte do microcontrolador.

```
case 'r':
  Wr(adc_read(0));
  break;
```

Se o carácter recebido não for nem o "s" nem o "r" não se executa nada.

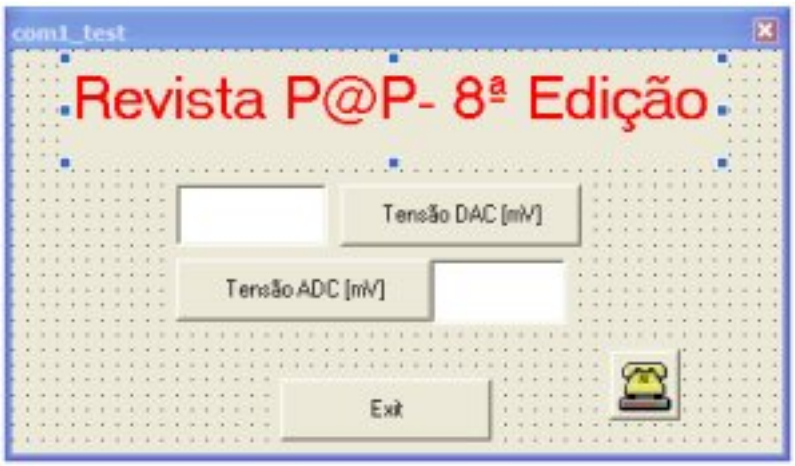
```
default:
  Putsr("\r parametro
incorrecto\r");
  break;
 }
}
}
}
```

Assim se termina o programa para o microcontrolador.

Interface em Visual Basic

Daqui em diante assume-se que o leitor está minimamente familiarizado com o programa Visual Basic

Para a interface pode-se começar por criar o layout da mesma:



Um dos componentes mais importantes desta interface é o componente MSComm. É aqui que se define como irá ser usada a porta série do PC. Também se adicionam duas caixas de texto e dois botões.

Assim que o programa for executado configura-se a porta série 1. Estes parâmetros têm que estar de acordo com os dados da inicialização da USART do microcontrolador.

```
Private Sub Form_Load()
  ' chose correct settings for your
device
  Settings = "115200,N,8,1"
  MSComm1.Settings = Settings

  ''''''''''''''''''''''''''''''
  ' Open COM1
  MSComm1.CommPort = 1
  MSComm1.OutBufferSize = 4
  MSComm1.PortOpen = True

  ' Make sure DTR line is low to
prevent Stamp reset
  MSComm1.DTREnable = False
  MSComm1.RThreshold = 1
  MSComm1.InputLen = 1

End Sub
```

É possível tornar esta interface mais robusta, para que perante situações não previstas esta reaja de uma forma esperada.

A caixa de texto referente à tensão da DAC tem o nome "tensão_dac" e o botão tem o nome "send_buttom".

Assim quando o botão "send_buttom" for pressionado será enviado o valor da caixa de texto para o microcontrolador.

Para não complicar a compreensão do código e para usar os recursos disponíveis no VB optou-se por enviar o valor da tensão num conjunto de 3 bytes, antecedidos do carácter "s".

```
Private Sub send_Click()
  MSComm1.Output = "s"
  MSComm1.Output =
Hex(tensao_dac.Text)
End Sub
```

Para se receber a tensão da ADC do microcontrolador usa-se o botão que tem a inscrição "Tensão ADC [mV]" . A este botão foi dado o nome "Request_buttom". Assim quando este for pressionado será enviado o carácter "r"

```
Private Sub request_Click()
  MSComm1.Output = "r"
End Sub
```

Assim que o microcontrolador receber o carácter "r", vai activar a ADC após a conversão envia o valor para a sua USART. Assim que o microcontrolador tiver enviado o valor da conversão pela USART, a porta série do PC receberá esse valor

```
Private Sub MSComm1_OnComm()
Dim sData As String
Dim Lido As Integer

  ' If Rx Event then get data
  ' and process

  If MSComm1.CommEvent = comEvReceive Then
    sData = MSComm1.Input

    tensao_adc.Text = Round(
                  Asc(sData) * 19, 4)
  End If

End Sub
```

A ADC fez uma conversão de 8 bits. O resultado dessa conversão estará compreendido entre 0 e 255, o que implica que este valor tenha que ser adaptado à natureza da conversão. Daí se multiplicar o valor recebido por 19.4. Para terminar a interface é conveniente que se desactive a porta série para que não existam conflitos futuros... Para testar tanto o envio de informação para a DAC pode-se ligar a ADC do microcontrolador à saída da DAC. A não coincidência da tensão lida da ADC com a tensão de saída da DAC está relacionada com o facto da ADC ser de 8 bits.

Ricardo Rocha

GOSTAS DE PROGRAMAÇÃO?

É DIFÍCIL ENCONTRARES TUTORIAIS EM PORTUGUÊS DE PROGRAMAÇÃO?

SENTES FALTA DE LER O QUE TE INTERESSA?

REVISTA

PROGRAMAR

acede já a

www.revista-programar.info

e vira uma página na tua vida

um projecto

portugal-a-programar.org

Programação em C 2ª Parte

Depois de uma primeira parte na edição anterior, este artigo é concluído nesta edição.

6 – Expressões Condicionais e Ciclos

Continuando na lógica. A lógica na programação desempenha um papel muito importante porque nos permite tomar decisões mediante de certas condições ou expressões condicionais. Nesta sexta secção vamos ver ciclos. Um ciclo é algo que tem uma condição de paragem e se ela for verdadeira [tiver um valor diferente de zero], o corpo do ciclo é executado, caso contrário não o é. À volta desta definição, existem três ciclos em C: while (), for(; ;) e do while (). Vejamos o primeiro.

O while (), é muito fiel à definição apresentada acima. Ou seja, tem uma condição e um corpo. Se a sua condição for verdadeira, o seu corpo é executado, caso seja falso, o corpo não é executado. Mas podemos pensar: isto é igual a um if (). Ora é igual nesta definição, mas tem uma particularidade que o torna muito interessante e útil. Um ciclo, e o próprio nome o indica, não pára de executar o seu corpo enquanto a condição expressa na condição do while (), for falsa, ou seja, todo o corpo do while () é executado enquanto ela for verdadeira. Vejamos um exemplo:

```
#include <stdio.h>
int main()
{
 int entrar = 1;
 int contador = 1;
 while (entrar == 1) // o mesmo que
 // while(entrar), visto que entrar=1
 {
  if(contador < 6)
  {
   printf("contador vai em:
                    %d\n",contador);
   contador++;
 } else {
   printf("contador chegou a
                    %d\n", contador);
   entrar = 0; // obriga ciclo a
   //terminar da próxima vez que testar
  }
 }
 return 0;
}
```

Ou seja, temos uma variável que condiciona o ciclo, entrar, que é inicializada com o valor 1 [verdadeiro], de modo a que possa entrar no corpo do ciclo while () e a variável contador com o valor 1, que vai sendo incrementada, de cada vez que entra no corpo do while () e que for menor do que 6 [ditado pelo if ()] ao mesmo tempo. Quando o contador for 5, ele entrará no ciclo uma vez mais, será menor do que 6 uma última vez e ao reentrar no ciclo pela última vez, a expressão condicional contador < 6 [6 < 6], dará falso e portanto segue para o else, que dirá o valor guardado no contador e porá a variável que condiciona a entrada no ciclo while (), a zero, para que da próxima vez que a sua condição seja testada, dê falso e o ciclo termine.

O segundo ciclo que vamos ver é o for (; ;). Tem esta forma sintáctica estranha mas vai ser fácil perceber. Como se vê, o for (; ;), está dividido em três partes. A primeira é dedicada à declaração de variáveis, a segunda às expressões condicionais do ciclo e a terceira aos procedimentos de incrementação ou de mudança de variáveis. Para explicar melhor, nada melhor do que um exemplo:

```
#include <stdio.h>
int main()
{
 int entrar = 1;
 for (int contador = 1; contador < 6;
                              contador++)
 {
   printf("contador vai em:
                    %d\n", contador);
 }
 printf("contador chegou a
                    %d\n", contador);
 return 0;
}
```

Neste ciclo, é possível repararem que fica tudo mais arrumado e simples de perceber. Temos a primeira parte que declara e atribui um valor à variável contador [não é obrigatório ser preenchido], depois temos a segunda parte que tem a condição de paragem do ciclo e na terceira parte temos o nosso modificador da variável, que neste caso a incrementa uma unidade, e enquanto contador < 6, o corpo do for (; ;) é avaliado e executado.

Por fim temos o ciclo do while (), que é praticamente igual ao while (), mas tem a particularidade de executar o seu corpo antes de testar a condição de paragem. Isto tem uma vantagem. Se quisermos que uma dada expressão seja executada independentemente de a expressão condicional do while () seja falsa ou não, ela é sempre executada, pelo menos, uma vez. E mais um exemplo:

```
#include <stdio.h>
int main()
{
 int entrar = 1;
 int contador = 1;
 do
 {
   printf("contador vai em:
                    %d\n", contador);
   contador++;
 }
 while (contador < 6);
  printf("contador chegou a
                    %d\n", contador);
 return 0;
}
```

Como se vê, o corpo do do while(), é executado antes da sua condição de paragem. Dependendo da situação, das variáveis em questão e do seu valor actual, dá jeito usar um tipo de ciclo que resolva melhor o problema. Com variáveis, expressões condicionais e ciclos, já é possível fazer muita coisa em C, é uma questão de se praticar e de se tentar descobrir melhor cada elemento aqui visto.

7 – Strings [cadeias de caracteres]

Até agora vimos vários tipos de dados conhecidos. A maior parte deles, através da matemática [int, float, double] e vimos um tipo de dados, char, que ajudará a perceber o tipo de dados que vamos ver nesta secção, strings. Uma string é um conjunto de caracteres seguidos que termina com o caracter especial, terminador de strings, '\0' [leia-se, barra zero]. Ou seja, uma string não é mais que uma frase que tem como limites, inicial e final, o primeiro caracter e o caracter especial '\0', respectivamente. Se quiséssemos escrever a string "olá", esta era composta da seguinte maneira:

```
'o' 'l' 'á' '\0' = "olá"
```

Em que a posição 0 [zero] da string tem o caracter 'o', a posição 1 tem o caracter 'l', a posição 2 tem o caracter 'á' e a posição 3 tem o caracter especial '\0'. Se repararmos, existem 3 caracteres visíveis na string, mas esta é formada por 4, uma vez que possui o caracter terminador de string '\0'. Outro aspecto importante tem a ver com o facto de a contagem e posterior selecção das posições começa com o número 0 [zero] e não no 1. Funciona como index, ou como posições de casas. Se a string tem n caracteres, as suas posições vão de 0 até n-1, e a posição n, é preenchida com o caracter terminador de string '\0'.

Como vimos até agora, quando queremos usar uma variável, temos de a declarar. Ora para criar uma variável do tipo string só temos de a declarar e de indicar o espaço que ela vai ocupar. Por exemplo, se a string em questão usar 20 caracteres, podemos declarar:

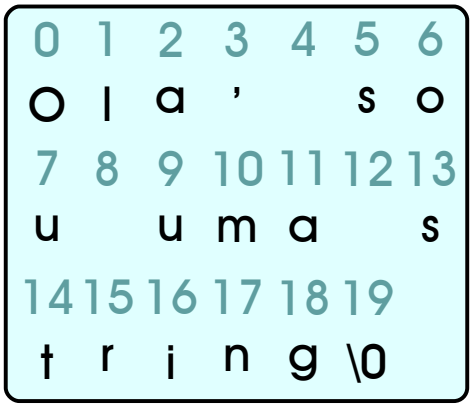
```
char a-minha-string[20];
```

E temos a nossa string com capacidade para 20 caracteres criada.

Em seguida, poderíamos usar o espaço reservado para ela da seguinte maneira:

```
a-minha-string = "Ola, sou uma string";
```

E internamente isto seria equivalente a:



São 20 caracteres dentro de uma string que contem 19 caracteres visíveis mais o caracter terminador '\0', totalizando 20 posições, indexadas de 0-19.

A manipulação de strings dá muito jeito e é bastante fácil. Vejamos: se quiséssemos aceder à posição número 3 da string [que nos dá um caracter], faríamos:

```
a-minha-string[3]
```

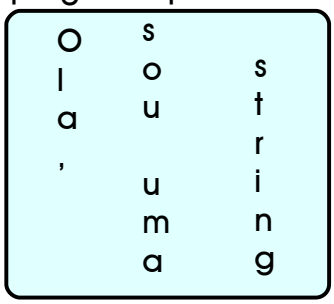
E para qualquer outra posição faríamos a-minha-string[n].

Um exemplo utilizando um simples ciclo, mostra como poderíamos imprimir um a um e separados por linhas, cada um dos caracteres da string acima criada e o respectivo número deles.

```
#include <stdio.h>
int main()
{
  int count;
  char myStr[20] = "Ola, sou uma
string";
  for(count = 0; myStr[count] !=
'\0'; count++)
  {
    printf("%c\n", myStr[count]);
  }
  return 0;
}
```

Ou seja, declarámos uma variável "count" do tipo inteiro sem nenhum valor à partida, e voltámos a declarar a nossa string, desta feita com o nome "myStr" , com as mesmas 20 posições e a mesma string "Ola, sou uma string".

Este programa produziria o seguinte:



Poderíamos ainda mostrar o número de caracteres imprimidos no ecrã acrescentado ao código o seguinte:

```
printf("numero de caracteres
imprimidos no ecrã : %d\n", count);
```

Antes do "return 0;" [fim do programa] e depois do loop for.

8 - Vectores

Como vimos na secção anterior, uma string é uma cadeia de caracteres todos ordenados e seguidos de um caracter especial terminador '\0'.
Ora um vector tem exactamente o mesmo conceito mas contém quaisquer elementos, dependendo do tipo de dados ao qual se associa o vector.
Por exemplo, poderíamos querer um vector de inteiros com os anos de várias datas de nascimento, assim faríamos:

```
int vecAnos[10];
```

Em que criaríamos e por consequência reservaríamos memória para 10 blocos de inteiros seguidos para o nosso vector. Para manipular o vector, faríamos tal e qual nas strings, uma vez que cada posição do vector tem um índice associado, que vai desde 0 até ao seu tamanho, n, menos 1 [0 -> n-1].

Utilizando o vector acima criado, modifiquemos, ou neste caso, preenchamos o seu conteúdo, uma vez que até agora o vector está vazio.

```
vecAnos[0] = 1987;
vecAnos[1] = 1988;
vecAnos[2] = 1989;
```

e por aí fora. Seria uma maçada andarmos a declarar todas as datas, se, para além do mais, elas fossem seguidas. Para isso usamos um simples algoritmo, para preencher o vector, sabendo o seu tamanho final.

```
#include <stdio.h>
int main()
{
  int vecAnos[10];
  int start = 1987;
  for(i = 0; i <= 10; i++, start++)
  {
    vecAnos[i] = start;
  }
  for(i = 0; i <= 10; i++)
  {
    printf("vecAnos[%d]=%d\n",
                        i,vecAnos[i]);
  }
  return 0;
}
```

Portanto o conceito de vector é o mesmo da string só que funciona com múltiplos tipos de dados pelo que ao vector de char chamamos-lhe string.

Apesar de tudo, através do que foi aqui explicado até à secção 8 dos vectores, é possível fazer-se muita coisa em C. No entanto, existe uma "matéria" bastante importante para a eficiência e organização de código nos programas: funções.

9 - Funções

Não é nenhuma novidade falar em funções visto que temos vindo a falar nelas e a usá-las desde o início. A mais falada, a mais usada e a essencial para que qualquer programa em C corra é a função main(). Mas em geral uma função apresenta a seguinte sintaxe:

```
tipo_de_retorno Nome_da_função (
               parâmetros declarados )
{
  CORPO;
  RETORNO;
}
```

Muito simples. Um pequeno exemplo tornará mais clara a ideia de função:

```
#include <stdio.h>

int func_soma(int x, int y)
/*função que retorna um int, que tem
o nome func_soma, que tem os
parâmetros declarados "int x, int
y", que tem um CORPO e um RETORNO do
inteiro relativo -> soma (x + y) */
{
  return (x + y); // ou int temp = x
+ y; seguido de return temp;
}

int main()
/* função que retorna um int, que
tem o nome main, que não tem
parâmetros declarados, que tem um
CORPO e um RETORNO do inteiro 0
[zero] *\
{
  int a, b, soma;
  printf("SOMADOR:\n");
  printf("Insira dois valores para\n");
  printf("a:");
  scanf("%d\n", &a);
  printf("b:");
  scanf("%d\n", &b);
  soma = func_soma(a, b);
  printf("A soma de %d com %d e
                  %d\n", a, b, soma);

  return 0;
}
```

Como podemos ver, a função "func_soma" criada para auxiliar a soma, tem o mesmo aspecto da função main() mas utiliza parâmetros declarados à partida. Ou seja, a função "func_soma", requer que se "passem" certos argumentos para ela funcionar. Neste caso, a função exige que passemos dois inteiros. É de notar que os nomes 'x' e 'y' poderiam ser 'a' e 'b' como foram enviados na chamada dentro da função main(), uma vez que são zonas distintas. Apenas foram escolhidos x e y diferentes de a e b para facilitar a percepção.

10 - Ponteiros

Geralmente, os ponteiros são a "dor de cabeça" dos aprendizes iniciados, e por isso costumam ser explicados no fim. Não vamos fugir à regra e por isso fica aqui a sua explicação. Embora possa parecer complicado de início, veremos que é muito simples, mas que apenas com prática sobre o assunto se consegue perceber bem o seu funcionamento.

Como vimos nas secções 7 e 8, strings e vectores, respectivamente, o seu tipo de dados é uma estrutura em cadeia em que podemos imaginar várias casas seguidas com conteúdos lá dentro respectivos ao seu tipo de dados. Os vectores serão deste tipo com valores inteiros dentro das casas e as strings serão deste tipo com caracteres dentro das casas.

No entanto, quando criamos uma string ou um vector, atribuímos-lhe um nome. É aqui que entram os ponteiros. O nome de um vector não é mais do que um ponteiro para a primeira casa [índice 0] do próprio vector. E um ponteiro não é mais do que um endereço da estrutura para a qual aponta.

Se criarmos a seguinte string:

```
char revista[9] = "PROGRAMAR";
```

8000H	8001H	8002H	8003H	8004H
P	R	O	G	R
0	1	2	3	4
8005H	8006H	8007H	8008H	8009H
A	M	A	R	\0
5	6	7	8	9

Este seria o esquema na memória. E, como foi explicado anteriormente, o nome da estrutura, "revista" é um ponteiro para a primeira casa da estrutura, que neste caso é uma string. Portanto, "revista" aponta para a casa 0 [zero] que contém 'P' e que por sua vez terá um endereço. Supondo que a string começa no endereço 8000H, então, "revista" tem o endereço 8000H.

Como um carácter ocupa 8 bits [1 byte] na memória, os endereços saltam de um em um como se vê na tabela. Se mandarmos imprimir, carácter a carácter, esta string, fazemos algo como isto:

```
char revista[9] = "PROGRAMAR";
char c;

while (*revista != '\0')
// '\0' é o carácter terminador de
// strings
{
  printf("%c", *revista);
  revista++;
}
```

Ora aparece aqui um símbolo novo: '*'. O símbolo asterisco, significa "conteúdo". Como foi explicado antes, o nome de uma string/vector não é mais do que um ponteiro para o endereço da primeira casa deste. Como tal, o nome em si contém o endereço de memória, mas falta poder aceder ao seu conteúdo.

Podemos indexar, fazendo revista, com i pertencente entre 0 a 9, mas como o nome da string/vector aponta para a primeira casa, para se aceder ao conteúdo dessa posição de memória, apenas se mete o '*' antes do nome. E para qualquer nome de uma estrutura destas, ao pôr-mos um asterisco antes de seu nome, estamos a aceder ao conteúdo da qual esse nome aponta.

Como vimos na tabela acima e pegando no pedaço de código, no início, "revista" aponta para 8000H. Se mandarmos imprimir o seu conteúdo, através da operação *revista, será impresso o carácter 'P' porque é o conteúdo de 8000H.

Se agora incrementarmos "revista" [que é um endereço], este avança uma posição na memória e portanto fica com o endereço 8001H, que por sua vez, *revista, passa a conter 'R'. E assim sucessivamente até *revista em que vai, contiver '\0' e aí para porque acabou de imprimir toda a string.

Este processo é muito semelhante com vectores ou com outros tipos de dados nestas estruturas sequenciais na memória.

Recapitulando, um nome de uma estrutura destas é um ponteiro para ela e contém o endereço para onde aponta [a morada da casa]. E para se lhe aceder ao conteúdo, usa-se o símbolo '*' antes do nome.

Conclusão

Existem muitas mais "matérias" em C que podem ser faladas, mas para um início, é importante cativar o aprendiz com certas matérias mais simples de modo a que, se este decidir avançar com a aprendizagem a fundo da linguagem em questão, não desmotive por falta de certos conhecimentos básicos que o impeçam de avançar e de perceber outros conteúdos mais avançados.

Embora seja possível aprender-se a programar, através de livros, tutoriais, artigos, aulas, etc., é importante perceber que sem prática autónoma, é muito difícil adquirir experiência e prática na programação de qualquer linguagem. Então, um apelo importante a todos os programadores iniciantes, é de porem em prática exercícios seus e de tentarem perceber a linguagem por sua auto-recriação.

Miguel Wahnon

Apresentação do Ubuntu 7.04 "Feisty Fawn"

Já passaram 6 meses desde o lançamento da última versão do Ubuntu, o 6.10 "Edgy Eft" e o lançamento do Feisty Fawn foi no dia 19 de Abril.

Com este artigo pretendo fazer uma apresentação simples de algumas das novas funcionalidades desta nova versão.

À primeira vista

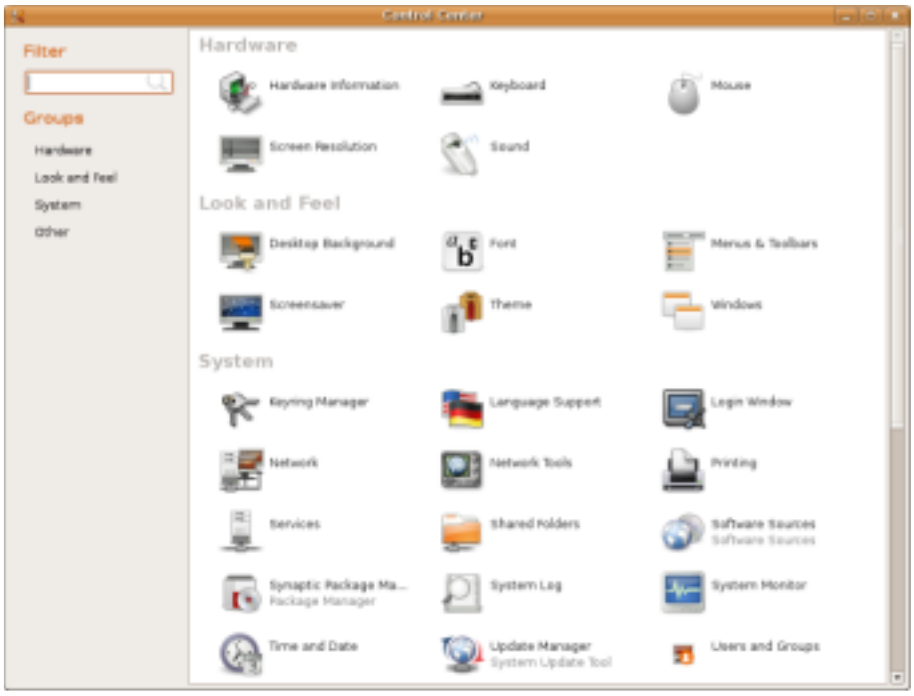
O aparência gráfica deste novo lançamento continua a seguir a aparência das versões anteriores com tons de castanho e laranja, havendo melhorias significativas.
Ao nível das aplicações instaladas, continua a ser um sistema completo, tendo aplicações para a maioria das actividades comuns – música, vídeo, documentação, instant messaging.

Desempenho

Podemos contar com algumas melhorias a nível de desempenho, algumas graças a novas versões de alguns sistemas, como o Xorg 7.2 (motor gráfico), Upstart 0.3.6 (Sistema de inicialização no Ubuntu desde o 6.10) e Kernel 2.6.20.
A inclusão do Xorg 7.2 ao Ubuntu foi uma surpresa, poucos estavam à espera da sua adição neste lançamento. Esta versão vem com várias melhorias significativas de estabilidade e funcionamento, como auto-configuração mais eficiente, suporte melhorado para gestores baseados em GL, como o Compiz ou Beryl e políticas de segurança mais extensas.

Gnome 2.18

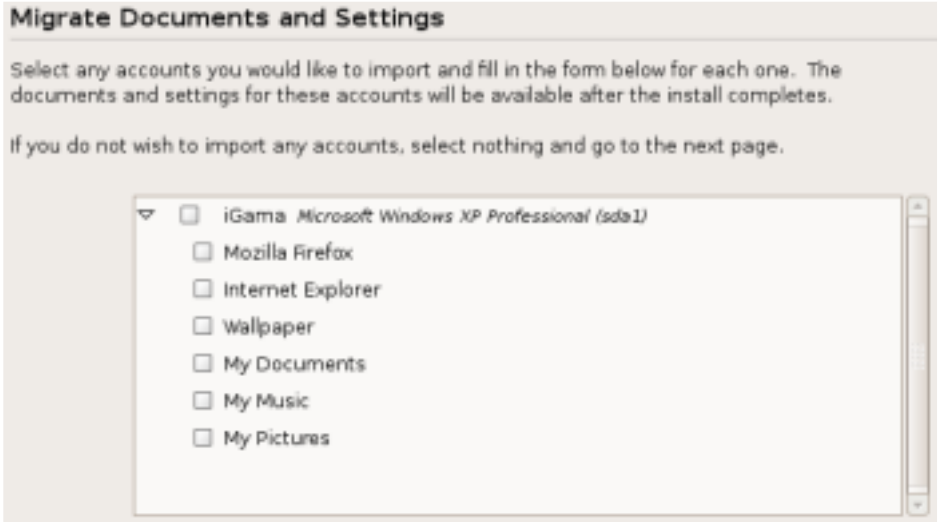
O novo Gnome vem com muitas melhorias a nível de velocidade e estabilidade, como uma interface melhorada, melhor suporte para rede, e muito mais. O GNOME Control Center, vem com uma nova interface que agrupa as funcionalidades em tarefas semelhantes. O Control center permite alterar várias definições no sistema, como fonts, teclado, som, temas, etc.



Instalação

A instalação gráfica está mais fácil, com novas melhorias tanto ao nível de estabilidade como funcionalidades do ubiquity - Sistema de instalação gráfico usado no Ubuntu.

Uma das novas funcionalidades do ubiquity é o "Migration-Assistant". Esta aplicação permite aos utilizares importarem documentos, ficheiros e definições de outros sistemas operativos que estejam instalados, e converter para os seus equivalentes em Ubuntu.
Por exemplo, pode copiar o conteúdo da pasta My Music no Windows para a pasta Music no Ubuntu, e adicionar ao Rhymtbox, ou se tiver o Firefox, IE ou Opera instalado, este irá compilar as bookmarks e adicionar as do Ubuntu.



Instalar Drivers Proprietárias para as tanto para as placas gráficas como wireless é mais simples, usando o Restricted Drivers Manager. Num clique os drivers são instalados.

Rede Simples

Seguindo a filosofia do Ubuntu para simplificar a experiência do desktop ao máximo de utilizadores, uma das especificações para este lançamento foi o "ZeroConfNetworking", que resumidamente é a capacidade de configurar o acesso à rede automaticamente e com o mínimo de intervenção do utilizador. O Sistema tentará configurar o que poder para tentar fazer a rede funcionar correctamente.

Uma das apicações que vem de base para ajudar nesta tarefa é o Network Manager, uma interface gráfica, no qual se pode configurar tanto a rede por ethernet, wireless e VPN.

Multimédia

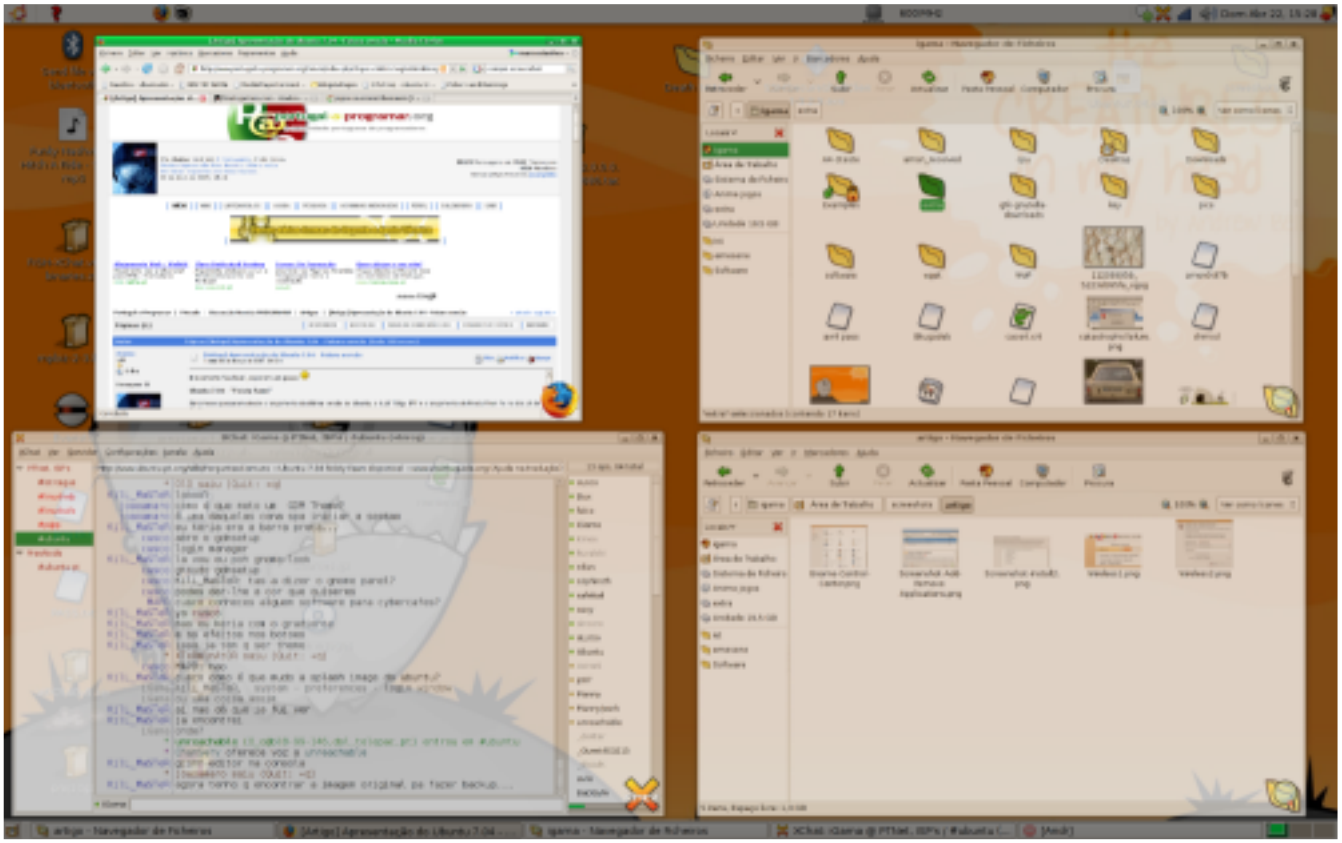
Um dos maiores desafios para um novo utilizador é a parte multimédia, isto é, a capacidade para reproduzir áudio e vídeo no seu Ubuntu. Mais uma vez, para simplificar, este processo já não é algo de outro mundo.
Ao executar um ficheiro media que não tenha nenhum codec compatível instalado, o sistema irá lançar um aviso e ajudar no processo de instalação dos codecs necessários.

Outra melhoria a este nível, foi a criação do pacote "Ubuntu-restricted-extras". Este pacote é na realidade um "meta-package" que aponta para a instalação de cerca de 20 pacotes para ajudar a experiência multimédia. Inclui Java JRE, Flash9, alguns codecs e mais uns pacotes.

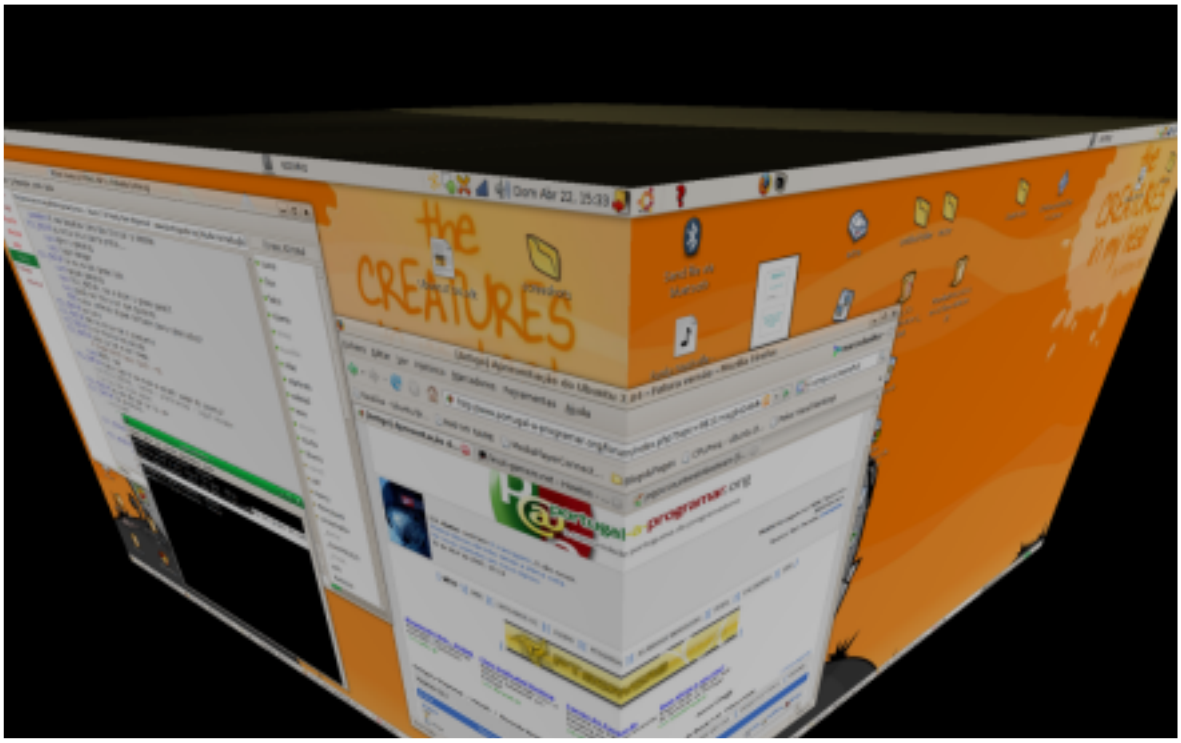
Efeitos Gráficos

Para os que pensam que os efeitos gráficos do Vista são novidade, deverão ficar admirados por saber que em Linux já existiam e não é preciso uma máquina última geração para funcionarem com fluidez.
O Ubuntu Feisty vem com o Compiz 0.3.6 instalado de base, e se a vossa gráfica for compatível, deverá funcionar logo sem nenhuma configuração extra.

Organização das janelas abertas



Efeito de "cubo"



Resumindo, este lançamento é a continuação da evolução visível no Ubuntu. A experiência do utilizador está mais simplificada e o sistema mais estável e compatível.

Aconselhado a todos os que querem experimentar GNU/Linux.

Moodle: Uma plataforma de eLearning

O que é o Moodle?

O Moodle é uma ferramenta que possibilita a gestão de aprendizagens e de trabalho colaborativo, permitindo a criação e gestão de cursos on-line em torno de comunidades com objectivos educativos. A ideia inicial foi de Martin Dougiamas, que em 1999 pensou criar o Moodle, sem qualquer objectivo comercial, pois acredita nas potencialidades do ensino e da construção do conhecimento através da Internet. Depois de algumas tentativas iniciais é em Agosto de 2002 que sai a versão 1.0 do Moodle. A partir daí a comunidade Moodle tem crescido bastante e são cada vez mais as escolas, universidades, centros de formação, empresas e organizações a utilizar esta ferramenta.

Em Portugal (que regista já um total de 858 "Moodles" registados) tem-se verificado uma grande adesão de várias escolas e centros de formação a esta ferramenta. Esta "moda" deve-se essencialmente à sua facilidade de uso, acompanhada da ausência de custos com o software, bem como ao novo Quadro de Referência da Formação Contínua de Professores na Área das TIC.

Questões técnicas

O Moodle é Open Source distribuído sob a GNU Public Licence. No final do mês de Março foi lançada a versão 1.8, melhorando a flexibilidade, a acessibilidade e a usuabilidade da plataforma, nomeadamente no que diz respeito à parametrização dos papéis dos utilizadores. O Moodle corre em PHP e suporta MYSQL ou PostgreSQL, embora nas últimas versões se possa ainda usar em alternativa Microsoft SQL Server ou Oracle.

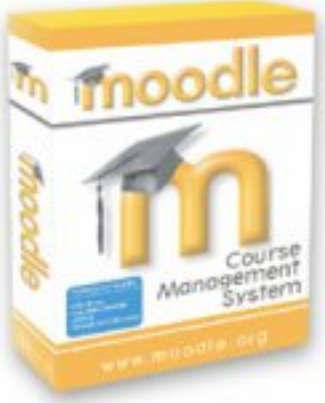
Para um bom funcionamento o Moodle precisa correr tarefas automatizadas (cron) e ter espaço disponível para a colocação dos vários recursos dos cursos a criar. No site oficial está ainda disponível um conjunto de plugins (módulos e blocos) que permitem alargar ainda mais as potencialidades do Moodle.

Potencialidades do Moodle

O Moodle permite a criação de cursos/disciplinas que podem ou não estar acessíveis a todos os visitantes. Nestes espaços podem depois ser criados e colocados recursos. O Moodle, tendo por base uma filosofia construtivista da educação, traz já um conjunto de ferramentas que facilitam a necessária interacção entre os participantes. Destaca-se aqui a possibilidade de criar Wiki's, fóruns, glossários, chats, blogs, bases de dados, diários, inquéritos, lições, testes. É tudo isto que pode tornar o Moodle num espaço de aprendizagem muito mais rico que um simples repositório de conteúdos. É esse o desafio lançado a quem pretende usar o Moodle em toda a sua plenitude.

Sites de referência

Página oficial do Moodle
http://www.moodle.org
Comunidade Portuguesa de Moodle
http://web.educom.pt/moodlept/
Página onde se pode testar o Moodle
http://www.opensourcecms.com



José Oliveira

O Blog da LinTeam

A LinTeam, uma das equipas do nosso forum que se dedica ao GNU/Linux, deu, durante o mês de Abril, um grande passo no seu trabalho: lançou um blog. Com este blog o objectivo é levar o Software Livre a cada vez mais utilizadores através de textos críticos, artigos de opinião, notícias e desenvolvimentos do mundo opensource.

http://www.linteam.portugal-a-programar.org

Estatísticas:
- 18 posts
- 45 comentários
- Mais de 1000 visitas únicas
- Mais de 8000 Hits
- Visitas de 16 países diferentes

Motji

O Mojiti permite personalizar vídeos alojados nos principais servidores de vídeo da Internet. Depois de escolhido o vídeo podemos colocar legendas, anotações, várias formas, imagens, animações,... tudo isto no local desejado. O Mojiti adiciona depois um layer com estas personalizações.

Recentemente foram ainda adicionadas novas funcionalidades a esta ferramenta. É neste momento possível também adicionar áudio e vídeo a um outro vídeo! Um recurso a conhecer.

http://www.mojiti.com

Conversão de Vídeos

Todos nós gostamos de ir ao YouTube ou a outros sites do género e ver alguns vídeos. Muitos gostam também de fazer download dos mesmos para a posterioridade. O problema surge no fim do download: tenho um vídeo em .flv, logo tenho de fazer download de um plugin ou de um programa! Normalmente isto é uma grande maçada principalmente quando queremos mostrar o vídeo aos amigos e eles têm de instalar um programa. Com o vixy.net esse problema evaporou-se. É possível converter no próprio site esses vídeos .flv para AVI, MOV para Mac, MP4 para o iPod e para a PSP, 3GP para qualquer telemóvel e MP3, que apenas converte o aúdio do vídeo.

http://www.vixy.net

Pelo menos é mais bonito que a concorrência...

Aí está ele...
...o BlueScreen of Death!

Será que a tortura resulta?

Chegámos ao mês de Maio, um mês que me costuma trazer muitas e boas lembranças. Porquê? Ora, foi em Maio de 2005 que a comunidade Portugal-a-Programar foi criada, mais precisamente no dia 28, às 19 horas e 18 minutos. Ou seja, fazemos no dia 28 deste mês de Maio, dois anos de existência. E para comemorar esses dois anos de trabalho e dedicação que tantos frutos têm trazido, o staff do Portugal-a-Programar decidiu lançar um concurso: RoboCode em Java. Para mais informações e para se inscreverem, sugiro que visitem a nossa secção Java ou que entrem em contacto directo com o moderador Knitter.

Falando do que se fez e do que se fará, estamos a planear avançar com outro projecto liderado e elaborado pelo staff já a partir de Julho. Terá de ser adiado para essa época devido aos exames e à importância que os mesmos terão na carreira profissional de cada um. Outros projectos têm surgido por iniciativa dos utilizadores e provavelmente veremos mais desenvolvimentos nas próximas semanas. Quanto ao que se fez, temos várias equipas a trabalhar em simultâneo em várias áreas - equipas de tradução, de desenvolvimento de software e de trabalho para a comunidade - e portanto é bem provável que muito em breve comecem a surgir notícias de projectos acabados. Desde o lançamento da última edição foram criadas duas equipas de tradução, uma liderada pelo Knitter outra pelo djthyrax, a Design Team, que terá como funções fornecer elementos de design às várias plataformas da comunidade.

Relembro que as candidaturas a estas equipas e a todas as outras estão abertas, pelo que todos os membros da comunidade se podem inscrever nas mesmas. Para o fazer, basta falar com um moderador, administrador ou com o gestor do projecto.

Em termos estatísticos, a comunidade continua em grande crescimento: no passado mês de Abril, tivemos 22192 vistantes únicos, com uma média de 1171 por dia. São números que indicam um bom crescimento e que nos motivam para continuar a trabalhar na expansão da comunidade.

Espero que tenham um bom mês de Maio, que acedam muito frequentemente ao fórum, que participem nas nossas discussões e que contribuam activamente para a comunidade. Vemo-nos no P@P.

Rui Maia

Queres participar na revista PROGRAMAR? Queres integrar este projecto, escrever artigos e ajudar a tornar esta revista num marco da programação nacional?

Vai a

www.revista-programar.info

para mais informação como participar
ou então contacta-nos por

revistaprogramar
@portugal-a-programar.org

Precisamos do apoio de todos para tornar este projecto ainda maior...

contamos com a tua ajuda

Equipa PROGRAMAR

Um projecto Portugal-a-Programar.org