

Prog1Box: Um Ambiente Linux sem Distratores para o Aprendizado de Programação

Matheus Gaudencio, Dalton S. Guerrero

¹Universidade Federal de Campina Grande
Laboratório de Práticas de Software
Av. Aprígio Veloso, s/n, Prédio SPLab, Bodocongó
58.429-900, Campina Grande, PB

matheusgr@copin.ufcg.edu.br, dalton@dsc.ufcg.edu.br

Abstract. *The computer system that students use to program and to do their exams is an important resource of programming courses. Such system should provide the best tools available, and remove distractions and any other barriers to the student activities. In this work we present a computer system that provides a complete Linux operational system with programming tools and web access. We provide an environment that has network controlled access and forbids system access to students. By doing so we try to reduce student distractions and even cheating behavior. We also provide an easy way to teachers install and adapt the student environment according their classroom needs. Moreover, this solution also avoids major changes of a computer laboratory IT infrastructure.*

Resumo. *O ambiente computacional em que alunos de programação fazem atividades é parte do ferramental básico de cursos de programação. Um ambiente apropriado deve dar ao estudante acesso às ferramentas adequadas para seu estudo e deve evitar distratores para a realização das suas atividades. Este trabalho apresenta um ambiente Linux que disponibiliza ao aluno as ferramentas necessárias para programar, dando acesso controlado a endereços Web. O ambiente impede o acesso a serviços que são fontes de distrações ou de comportamento indevido e permite que o professor rapidamente instale aplicativos e altere o conjunto de endereços acessíveis. Isto é possível sem causar grande impacto na estrutura de um laboratório de computação.*

1. Introdução

No ensino de programação, é comum que os alunos resolvam questões e façam atividades em que exercitem a prática de codificação. A infraestrutura computacional para a realização de tais atividades deve fornecer ferramentas básicas de programação, tais como editores, compiladores e ambientes integrados de desenvolvimento (IDE), podendo também usar alternativas online para isto. Como exemplo, no ensino de programação, é comum o uso de ferramentas de submissões online de programas para serem avaliados por testes automáticos implementados pela equipe de ensino [Gaudencio et al. 2013].

Entretanto, surge um problema ao oferecer ao estudante acesso irrestrito online: a presença de distratores e de elementos responsáveis pela interrupção das atividades de estudo, como ferramentas de *chat* [Wijekumar and Meidinger 2005]. Assim, o acesso irrestrito a serviços online pode ser mais um distrator do que um benefício para o aluno

durante atividades que façam uso de um computador. Se faz necessário que se controle quais as ferramentas online podem ser acessadas de forma tornar o aluno produtivo e focado nas atividades propostas.

Ainda, durante a realização de atividades práticas de avaliação em programação, apenas controlar o acesso a rede não é suficiente para se evitar a ocorrência de comportamento indevido. Especialmente quando se deseja oferecer um sistema operacional completo ao aluno, é importante que se tenha um ambiente que seja capaz de reduzir a possibilidade comportamento fraudulento [Sessink et al. 2004]. Além disso, este sistema não deve ser restrito o suficiente de forma que ele não possa ser alterado caso o professor deseje incluir novas ferramentas ou atividades durante uma aula ou avaliação.

É nesse contexto que o ambiente que os alunos fazem uso em sala deve ser capaz de prover as ferramentas básicas, e, ao mesmo tempo, ser capaz de dar acesso online restrito e controlado pelo professor de forma a evitar distratores ou possibilidades de fraude. O Prog1Box, descrito neste trabalho, busca oferecer um sistema operacional completo e com acesso online restrito, mas que pode ser facilmente adaptável para a instalação de novas ferramentas ou para o acesso de novos serviços online. A proposta apresentada pode ser completamente customizada pelos professores para atender a suas necessidades específicas.

2. Trabalhos Relacionados

O primeiro trabalho relacionado [Moskal et al. 2009], trata da preparação de um sistema operacional para a aplicação segura de avaliações. Neste ambiente, o único acesso a rede oferecido era para acesso a arquivos de um servidor local. Já uma segunda proposta relacionada [Basar and Genc 2006] disponibiliza acesso online a um único serviço Web através de um navegador que é alterado de forma evitar que o aluno possa utilizá-lo para acessar outros serviços não permitidos ou usar o sistema operacional que executa o navegador. Diferente desses trabalhos apresentados, o Prog1Box busca oferecer um sistema completo com acesso online de forma controlada pelo professor.

Existe um ambiente [Sessink et al. 2004] com uma proposta mais semelhante ao Prog1Box e que busca oferecer acesso a um ambiente Linux completo, mas com restrições de uso quanto aos aplicativos e aos endereços utilizados. Esta última abordagem também faz a restrição de endereços, o que se assemelha bastante a proposta deste trabalho, mas difere na facilidade em adaptar o ambiente para a instalação de novos aplicativos ou de novas regras de restrição.

3. Prog1Box

O Prog1Box deseja contemplar professores que fazem uso de um laboratório de computação para aulas práticas de uma disciplina, em especial, programação e também reduzir o esforço da atividade de administradores de sistema que precisam manter a infraestrutura de um laboratório de computação. São funcionalidades principais do Prog1Box:

1. Conter ferramentas essenciais como: Python, Geany (IDE), leitor de documentos (PDF), um navegador moderno além de possibilitar a instalação de novas ferramentas de acordo com a necessidade da aula;

2. Possibilitar a fácil liberação de endereços Web e permitir o monitoramento das máquinas online;
3. Oferecer a execução de ferramentas sem privilégios no cliente, ter armazenamento volátil e resistente a alteração, impedindo comportamento indevido.

Para que este ambiente possa ser facilmente utilizável num laboratório, é preciso que o mesmo apresente requisitos não funcionais básicos como tolerância a falhas de rede e dos serviços do ambiente bem como ser pequena e rapidamente inicializada. Tudo isto, sem a necessidade de máquinas robustas e sem precisar alterar a estrutura do laboratório. Ainda, para os alunos, o ambiente deve dispor de uma interface amigável, que possa ser usada de forma efetiva e eficiente e que satisfaça as necessidades destes estudantes.

3.1. Arquitetura

O Prog1Box é composto de 3 entidades básicas: *i*) O ambiente de programação, que corresponde ao sistema operacional Linux oferecido ao estudante com aplicações e um proxy de internet que controla o acesso a páginas; *ii*) Servidor Prog1Box, que localiza-se num endereço conhecido e oferece ao cliente Prog1Box endereços a serem liberados e comandos a serem executados para a instação de novos programas ou recursos na máquina, e; *iii*) o cliente Prog1Box, que executa dentro do ambiente e que se conecta ao servidor Prog1Box para receber comandos e endereços a serem liberados. Todos os arquivos de configuração para a geração de imagem, bem como código fonte do servidor e demais programas adaptados para a utilização do Prog1Box estão disponíveis em <https://github.com/matheusgr/prog1box>.

3.1.1. Ambiente

Para construir o ambiente de programação, utilizou-se o live-images¹, uma aplicação feita especialmente para criar imagens (ISO) de live CDs. Este programa permite, dado um conjunto de regras iniciais, a geração de um ambiente de programação com três diferentes formas de inicialização: *i*) CD/Pendrive; *ii*) inicialização de um arquivo do computador, e; *iii*) uso de uma máquina virtual. Enquanto a primeira estratégia adiciona uma proteção física, as demais alternativas oferecem comodidade ao administrador do laboratório por evitarem o uso de mídia física ou por tirar a necessidade de reiniciar a máquina.

Na última versão gerada pelo live-image, criou-se uma imagem linux utilizando o sistema Gnome (versão 3.4) que faz uso de um gerenciador de redes que automaticamente se conecta à rede local através do serviço de DHCP. Também foram instalados uma IDE Python (Geany), um leitor de PDF (evince) e um browser (Firefox). Foram removidas aplicações que poderiam ser utilizadas como distratores (jogos, editores de imagem) ou que não são inicialmente utilizadas no ambiente (gerenciador de impressão, bluetooth, programas de 'chat').

Foi adicionado o MITM proxy ao ambiente e este foi configurado de forma que os usuários tenham acesso apenas ao endereço e porta desse proxy local. As demais aplicações quando executadas como usuário privilegiado (root), como o próprio proxy, tem acesso irrestrito a rede. Este mesmo proxy permite controle ao acesso de páginas

¹Live systems project. <http://live-systems.org/devel/live-images/>

HTTPS e foi adaptado para permitir a configuração dinâmica dos endereços acessíveis. Além do MITM proxy, é instalado o cliente do Prog1Box. Este será responsável pelo controle remoto do ambiente. Ambas as aplicações são monitoradas localmente por um serviço de recuperação. Em caso de travamento ou fim de execução abrupta, ambos são reinicializados.

Todo este ambiente é executado com um disco que fica armazenado em memória RAM. Assim, este disco é limpo a cada inicialização da máquina. Isto representa uma vantagem, por permitir maior segurança do tipo de informação que será acessível, entretanto, pode dificultar a experiência do aluno em sala em casos de falha de sistema. Entretanto, caso o professor deseje, é possível fazer a instalação de aplicações que oferecem acesso temporário a um servidor permanente de dados.

3.1.2. Cliente e Servidor Prog1Box

O Cliente Prog1Box é uma aplicação que, a cada 60 segundos, acessa uma página HTTPS contendo um script a ser executado localmente. O responsável por esta página é o servidor Prog1Box que fica acessível num domínio bem conhecido (<https://prog1box.appspot.com/>) e protegido por uma certificação digital. No script padrão, são definidos os endereços acessíveis online bem como qualquer comando adicional, em geral executados para liberar algum recurso da imagem ou instalar novos programas.

O servidor Prog1Box pode servir diferentes scripts de acordo com o endereço IP de origem, tornando flexível a solução proposta para a utilização de diferentes ambientes ao mesmo tempo. O cliente, ao terminar de executar o script, faz uma nova requisição ao servidor Prog1Box desta vez passando a saída da execução do script para o servidor através de uma requisição POST. Este resultado pode ser exibido ao professor ou administrador do laboratório permitindo que o mesmo acompanhe (monitore) o funcionamento das máquinas no ambiente.

4. Avaliação e Uso

O ambiente vem sendo utilizado no laboratório de computação da Universidade Federal de Campina Grande para a disciplina de Laboratório de Programação 1 desde novembro de 2012. Ao total, mais de 300 alunos já fizeram uso da imagem em 15 diferentes turmas da disciplina. O ambiente atualmente é inicializado como boot alternativo ficando a imagem instalada na máquina do laboratório. Na última instalação, a ISO ocupa 432 Mb o que permite o seu funcionamento em CDs convencionais.

Para avaliação do Prog1Box, é analisado se todos os requisitos descritos foram atendidos. Os requisitos funcionais são atendidos pela forma como o ambiente foi construído. Para avaliar a eficiência em termos de tempo de inicialização e memória usada da imagem, a mesma foi testada em uma máquina virtual executada em um computador Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz. A máquina tinha acesso a 512 Mb de RAM, o que representa um computador de baixo custo. Foi testado o uso de memória real do sistema com duas aplicações abertas (geany/firefox). A memória usada foi de $209,1 \pm 2,2\text{Mb}$ (95% de confiança) e representa o total de memória usada eliminando o que existe em *cache* ou *buffer* (memória descartável). O tempo de inicialização obtido foi de $20,2 \pm 0,3\text{s}$ e leva em conta o momento que a máquina é inicializada até o primeiro

instante que toda a interface gráfica está pronta pra uso (incluindo tempo de inicialização da BIOS).

Para testar usabilidade, utilizou-se o questionário de usabilidade pós-uso para sistemas computacionais (CSUQ) [Lewis 1995]. Este questionário foi aplicado aos alunos de 2014.1 da Universidade Federal de Campina Grande. O questionário era de livre resposta e foi pedido aos 86 alunos do curso após a utilização do ambiente em 5 avaliações ao longo de 6 semanas. Ao todo foram obtidas 25 respostas que buscaram avaliar a usabilidade do sistema, a qualidade da interface e a qualidade geral do sistema. Os alunos também foram questionados se já tinham algum contato prévio com Linux e todo o questionário foi apresentado numa escala de Likert (de 1 a 7). O sistema foi avaliado com nota geral de $5,9 \pm 0,2$, sendo $6,0 \pm 0,2$ a nota para a usabilidade do sistema e $5,5 \pm 0,1$ a nota para a qualidade da interface. Estes resultados são bastante positivos considerando que os alunos questionados apresentavam pouca familiaridade com Linux $2,6 \pm 0,8$.

5. Conclusões

Este trabalho apresentou um ambiente computacional Linux customizado para o adoção em contextos de atividades de sala de aula de ensino de programação introdutória. O ambiente tem seu controle e acesso restritos de forma a evitar distratores em atividades de estudo, bem como para evitar potenciais fontes de fraude por parte dos alunos em atividades de avaliação. Como principal diferencial, ao manter uma estrutura de controle externo do ambiente, este passa a ser facilmente atualizável e adaptável para as necessidades dos professores em sala de aula. Ainda, sua arquitetura e construção permite um controle detalhado quanto a endereços acessados, e um bom resultado quanto ao seu desempenho, usabilidade, capacidade de manutenção e tolerância a falhas.

Referências

- Basar, E. and Genc, B. (2006). A software system for secure computer aided exams. In *Proceedings of the 2006 International Conference on Frontiers in Education: Computer Science & Computer Engineering, FECS 2006*, pages 24–30, Las Vegas, Nevada, USA.
- Gaudencio, M., Wanderley, L. F., Lemos, F. W., de Araújo, E. C., Figueiredo, J. C. A., and Guerrero, D. D. S. (2013). Eu sei o que vocês fizeram (agora e) na aula passada: o tstview no acompanhamento de exercícios de programação. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 24, Campinas, SP.
- Lewis, J. R. (1995). Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78.
- Moskal, P., Caldwell, R., and Ellis, T. (2009). Evolution of a computer-based testing laboratory. *Innovate: Journal of Online Education*, 5(6).
- Sessink, O., Beeftink, R., Tramper, J., and Hartog, R. (2004). Securing web-based exams. *J.UCS Journal of Universal Computer Science*, 10(2):145–157. http://www.jucs.org/jucs_10_2/securing_web_based_exams.
- Wijekumar, K. and Meidinger, P. (2005). Interrupted cognition in an undergraduate programming course. *Proceedings of the American Society for Information Science and Technology*, 42(1).