

ArchC Platform Manager - Um gerenciador de pacotes para plataformas de sistemas embarcados

Matheus Boy, Rodolfo Azevedo

¹Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6172 – 13083-852 – Campinas – SP – Brazil

matheus.boy@students.ic.unicamp.br, rodolfo@ic.unicamp.br

Abstract. *This paper presents the ArchC Platform Manager (ACPM) tool, created to help the management of project and developing of embedded systems. The ACPM lets you organize your local components, and helps obtaining new components through web repositories. The ACPM is the sucessor tool of ArchC Reference Platform (ARP), and the improvements brought in by the ACOM are the focus of this paper. The ACPM is available for download at the ArchC web-site.*

Resumo. *Este artigo apresenta a ferramenta ArchC Platform Manager (ACPM), criada para auxiliar o gerenciamento de projeto e desenvolvimento de sistemas embarcados. O ACPM permite organizar os componentes locais, além de facilitar a obtenção de novos componentes através de repositórios web. O ACPM é a ferramenta sucessora do ArchC Reference Platform (ARP), e as melhorias implementadas no ACPM são o foco deste artigo. O ACPM está disponível para download no site do ArchC.*

1. Introdução

O desenvolvimento e projeto de sistemas embarcados requer a criação de múltiplas versões do mesmo sistema, podendo variar desde alguns detalhes até grandes componentes. Gerenciar essas múltiplas versões não é algo trivial, e é parte crucial no desenvolvimento de novos sistemas embarcados. Pensando nisso, foi desenvolvido o ArchC Platform Manager (ACPM), sucessor do ArchC Reference Platform (ARP), ambas ferramentas desenvolvidas no Laboratório de Sistemas de Computação (LSC) do Instituto de Computação (IC) da Universidade Estadual de Campinas (Unicamp).

O ARP era um gerenciador de plataformas que consistia em uma estrutura de diretórios, um Makefile padrão do ARP e um software utilitário que permitia empacotar e desempacotar componentes.

O ACPM veio como um sucessor aprimorado do ARP, e sua expansão e aprimoramento são o foco da atividade de Iniciação Científica. Essencialmente, o ACPM é uma estrutura de diretórios associada a um software gerenciador de pacotes de plataformas de sistemas embarcados. A principal função do ACPM é organizar os componentes numa estrutura sistemática e possibilitar a obtenção de novos componentes para geração de plataformas. A obtenção de componentes se dá através dos mesmos já empacotados, através de download de repositório na web. Após o download, o software desempacota o componente, estando este pronto para uso. Essa é a principal diferença do ACPM para o ARP. O ACPM consegue obter novos componentes (pacotes) on-line, enquanto o ARP

não possui essa funcionalidade. Outras funcionalidades foram inseridas no ACPM, e outras que já existiam no ARP foram aprimoradas. Maiores detalhes serão dados mais adiante neste artigo.

Um trabalho relacionado ao ACPM é o ESLBench, também desenvolvido no LSC. O ESLBench é um benchmark para medir o desempenho de plataformas descritas em ArchC e SystemC, e utilizou diversos recursos do ACPM, como a estrutura de diretórios e os esquemas de Makefile próprios do ARP/ACPM. Mais sobre isso será dito neste artigo.

O artigo é organizado como a seguir. A seção 2 apresenta os conceitos básicos, a seção 3 apresenta a ferramenta ACPM, a seção 4 apresenta as melhorias feitas na ferramenta durante a atividade de Iniciação Científica, a seção 5 apresenta os resultados, a seção 6, as conclusões e a seção 7, os agradecimentos.

2. Conceitos básicos

Esta seção descreve os conceitos básicos necessários para o bom entendimento deste artigo. Uma definição que se faz importante no primeiro momento é o conceito de plataforma. No contexto do ACPM, uma plataforma é um conjunto de componentes de hardware que incluem, no mínimo, um processador e um periférico, preferencialmente tendo vários periféricos interligados e também algum mecanismo de interconexão entre eles (barramento, NoC, etc). A existência de um processador também garante que a plataforma seja programável e, portanto, possa ter uma finalidade mais geral, onde cada software diferente pode especializá-la para um uso distinto. Exemplos de plataformas podem ser vistos na Figura 1.

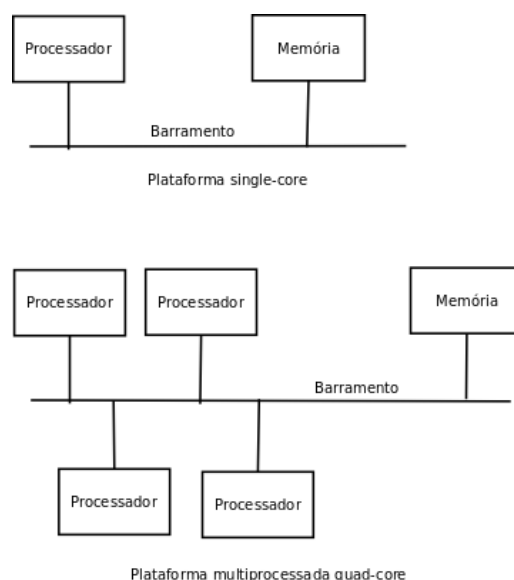


Figure 1. Exemplos de plataformas

2.1. ArchC

O ArchC é um Linguagem de Descrição de Arquitetura baseada no SystemC, desenvolvida pelo LSC, que permite a descrição em alto nível de um processador, além de oferecer a geração automática de simuladores e ferramentas de desenvolvimento de software.

Os simuladores gerados por ArchC podem ser utilizados tanto em sistemas isolados quanto em plataformas. O ACPM usa processadores descritos em ArchC.

2.2. SystemC

SystemC é um conjunto de classes C++ e macros que fornecem um kernel de simulação orientada a eventos em C++. SystemC permite a um projetista simular processos simultâneos, cada um descrito usando simples sintaxe C++. Processos SystemC podem comunicar-se em um ambiente em tempo real simulado, utilizando sinais de todos os tipos de dados oferecidos pelo C++, algumas entidades adicionais oferecidas pela biblioteca do SystemC, além de entidades definidas pelo usuário.

Todos os simuladores gerados por ArchC são codificados em SystemC e todos os componentes de plataformas também o são.

2.3. ArchC Reference Platform

O ARP é uma ferramenta de gerenciamento de pacotes para geração de plataformas de sistemas embarcados modelados em ArchC. Faz parte do projeto ArchC e foi desenvolvido no LSC.

O objetivo principal do ARP é organizar e facilitar a geração de plataformas. O ARP organiza os componentes da plataforma numa estrutura de diretórios própria de modo que seja possível gerar diversas configurações de plataforma sem gerar redundância entre os componentes. Desse modo, cada componente é compilado apenas uma vez e utilizado pela plataforma como uma biblioteca. Tanto os componentes quanto as plataformas são descritos em ArchC e em SystemC.

A estrutura de diretórios do ARP pode ser vista na Figura 2. Essa estrutura foi herdada pelo ACPM e será melhor explicada mais adiante neste artigo.

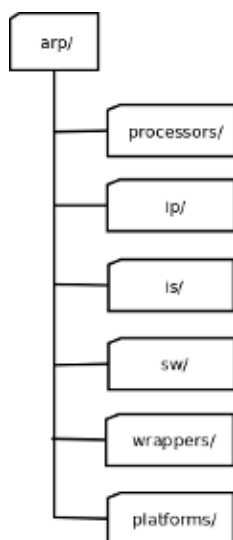


Figure 2. Estrutura de diretórios do ARP

2.4. ESLBench

O ESLBench é um benchmark de plataformas que está sendo desenvolvido no LSC. As plataformas podem utilizar processadores MIPS, PowerPC, SPARC e ARM descritos em

ArchC, além de periféricos descritos em SystemC comuns entre os componentes. A versão em desenvolvimento do ESLBench possui softwares dos benchmarks SPLASH-2, ParMiBench e MiBench. O ESLBench é o primeiro grande projeto do LSC a fazer uso do ACPM.

3. Visão geral da ferramenta ACPM

O ACPM organiza os componentes das plataformas em uma estrutura com 6 diretórios, e cada componente possui seu respectivo subdiretório. A estrutura de diretórios organiza os componentes nas seguintes categorias:

- processors: Diretório que contém os modelos de processadores.
- ip: Diretório que contém os IPs (Intellectual Properties) disponíveis. Cada IP colocado neste diretório se torna disponível para utilização em plataformas.
- is: Diretório que contém os elementos de interconexão disponíveis para a construção de plataformas, como barramentos, roteadores, NoCs, etc.
- sw: Diretório que contém os softwares que serão rodados nas plataformas. Um mesmo software pode ser executado em mais de uma plataforma, embora isso não seja muito comum.
- wrappers: Diretório que contém os conversores de protocolos que podem ser utilizados para conectar componentes distintos das plataformas.
- platforms: Diretório que contém as plataformas desenvolvidas. Cada versão diferente da plataforma deve possuir um diretório diferente, facilitando o gerenciamento das mesmas.

Essa estrutura é herança direta do ARP. Vale ressaltar que a classificação e separação dos componentes é o que permite a reutilização inteligente dos mesmos. Por exemplo, se existem duas plataformas, uma multicore e outra singlecore, cada uma terá seu respectivo diretório dentro de platforms, e cada plataforma instanciará o número apropriado de processadores no seu respectivo código fonte, assim como os demais componentes (IPs, barramentos, wrappers).

Ao se realizar o download da ferramenta é necessário iniciá-la através de um comando específico. Feito isso, o ACPM cria a estrutura básica de diretórios, cria dois arquivos de configuração, um para os endereços de repositórios web e outro para o repositório local, e baixa o Makefile. Assim, a ferramenta já está pronta para uso. O software, em linhas gerais, permite:

- obtenção de novos componentes em um repositório web;
- empacotamento de componentes desenvolvidos;
- listagem de componentes disponíveis localmente;
- listagem de componentes disponíveis no repositório web;
- geração de templates;
- suporte a múltiplos repositórios web;
- submissão de pacotes para um repositório.

Essas são as principais funcionalidades atualmente presentes no ACPM, contudo, isso não quer dizer que elas sempre fizeram parte da ferramenta, ou que sempre funcionaram da mesma forma. Todas sofreram modificações decorrentes do trabalho desenvolvido na atividade de Iniciação Científica, e a ferramenta como um todo foi amplamente expandida e melhorada.

4. Melhorias e expansões da ferramenta

Inicialmente, o trabalho de melhoria do ACPM se deu através do estudo e análise do código original do gerenciador do ARP. Após devida compreensão e detecção de pontos de interesse para melhoria ou modificação no código, iniciaram-se os trabalhos de modificação no gerenciador.

Um ponto interessante notado desde o princípio foi que o gerenciador da ARP foi originalmente implementado em Python, e escolheu-se manter essa como ferramenta de implementação do ACPM, devido à praticidade dessa linguagem para implementações rápidas. Outra característica boa do Python é a vasta biblioteca padrão da linguagem, facilitando tarefas como geração de arquivos zip, chamadas para o interpretador de comandos do sistema operacional e acesso à web.

Após extenso estudo do código do gerenciador, foram detectados muitos pontos passíveis de melhoria. Existia um tratamento de exceções muito superficial, tornando o código pouco tolerante a falhas. Na maior parte das vezes, quando era lançada uma exceção, eram emitidas verbosas mensagens de erro do interpretador do Python, que não davam pistas sobre o real problema que gerou a exceção. Por exemplo, apesar de ser um gerenciador de pacotes que buscava recursos na web, nenhum método que buscava dados on-line estava dentro de um bloco de tratamento de exceções. Além disso, nenhuma outra operação de I/O estava com suas exceções devidamente tratadas.

Depois de detectados esses pontos de interesse, foi iniciado um minucioso e extensivo trabalho de blindagem do código. Todos os pontos passíveis de exceção foram inseridos em blocos try/except, inclusive diferenciando-se alguns tipos específicos de exceções de modo a receberem tratamento apropriado ao invés de um tratamento genérico e inespecífico. O processo de blindagem do código foi gradual e iterativo. Inicialmente, o tratamento de exceções se resumia a emitir uma mensagem mais clara ao usuário do que uma mera exceção do interpretador do Python. Posteriormente, o tratamento de exceções foi evoluindo de modo que o gerenciador continuava sua execução mesmo com o surgimento de uma exceção. Depois desse processo, a versão original do software gerenciador foi declarada como blindada e tolerante a falhas, além de estar apta a receber novas funcionalidades e outras melhorias no código.

Além das correções quanto ao tratamento de exceções, foram corrigidos alguns aspectos do código original do gerenciador. Algumas funções redefiniam funções builtin do Python, como list(), mas sem expandir o funcionamento dessa função. Desse modo, os nomes das funções foram trocados de modo que nada da linguagem fosse redefinido.

A primeira funcionalidade a ser incluída foi o suporte a múltiplos repositórios. Isso foi implementado de modo que sempre que o gerenciador for rodado ele seleciona o primeiro repositório válido no arquivo de lista de repositórios web. Esse arquivo é inicialmente gerado quando se inicia o ACPM após o download do mesmo, e vem com apenas o endereço do repositório padrão. Esse arquivo é facilmente editável pelo usuário, podendo acrescentar outros repositórios conforme lhe aprouver. O arquivo funciona como uma fila de prioridades. Portanto, o repositório que estiver na primeira linha possui a maior prioridade, na segunda linha, a segunda maior prioridade e assim sucessivamente. A seleção automática de repositório tem um funcionamento relativamente simples. Essencialmente, o gerenciador tenta acessar sequencialmente cada repositório listado no arquivo, e se-

leciona o primeiro que retornar uma resposta HTTP 200 (OK). Se nenhum repositório retornar uma HTTP 200, o repositório padrão, hardcoded, é selecionado.

Depois disso, foi proposta a mudança do formato de empacotamento dos componentes de tar para zip. Isso se deu devido ao fato do gerenciador do ARP originalmente gerar os pacotes chamando tar(1) num subprocesso aberto pelo interpretador de comandos do sistema operacional. Essa decisão, contudo, restringe o uso do gerenciador a um ambiente POSIX. Como o Python suporta nativamente arquivos zip através de uma biblioteca padrão, a modificação dessa funcionalidade foi relativamente simples. Foi criada uma classe auxiliar para facilitar o manuseio de tais arquivos, e a inicialização do gerenciador após o download do mesmo precisou ser alterada. Isso se deu devido ao fato de originalmente ser realizado o download de um arquivo tar com a estrutura básica de diretórios durante a inicialização do ACPM, ao passo que o formato zip não permite explicitamente a compactação de uma árvore de diretórios. O que se faz para comprimir uma estrutura de diretórios usando zip é renomear os arquivos de modo a concatenar o path de diretórios do arquivo com o próprio nome do arquivo. Contudo, isso não permite adicionar diretórios vazios a um arquivo zip, de modo que a estrutura básica do ARP precisaria ser gerada de outra forma na inicialização do gerenciador. Isso foi resolvido de modo que os diretórios são criados dinamicamente na função de inicialização do gerenciador, através da biblioteca padrão do Python. Resolvida essa questão, foi modificada a funcionalidade de empacotamento de componentes de modo a comprimir os arquivos para zip ao invés de tar.

Outra funcionalidade proposta foi a implementação de um parser de YAML no próprio código do gerenciador. O ACPM usa YAML na listagem de componentes disponíveis nos repositórios web. YAML é um formato de serialização de codificação de dados facilmente legível e editável por seres humanos. Originalmente, o parsing era feito utilizando-se uma biblioteca desenvolvida pelo próprio projeto YAML, contudo, não padrão do Python. Sendo assim, para o gerenciador funcionar de forma correta, era necessária a instalação dessa biblioteca, algo não-trivial num ambiente não-POSIX. A implementação do parser de YAML foi uma tarefa não muito trivial, visto que seu principal desafio não residia no conhecimento da biblioteca padrão do Python, mas sim na capacidade de projeto e desenvolvimento de algoritmos por parte do aluno. Sendo assim, foi um dos problemas mais interessantes resolvidos no ACPM. Essencialmente, lê-se o arquivo YAML e monta-se uma hashtable (dicionário) a partir do arquivo. O parser foi implementado com sucesso e é parte essencial da atual versão do ACPM. Laboratório de Projetos de Sistemas Computacionais

Implementou-se uma forma inicial de um “put”, ou seja, submissão de pacotes para um repositório. A forma inicial não permite, explicitamente, a submissão de componentes para um repositório web, embora o usuário possa fazer com que essa funcionalidade se comporte assim. Essencialmente, o put foi implementado com uma chamada para a função empacotadora passando como argumento um componente específico, e a subsequente cópia do componente empacotado para um determinado diretório. O diretório-alvo do put pode ser alterado através de um arquivo de configuração, gerado automaticamente pelo ACPM, mas de fácil edição, de modo que o usuário possa especificar um local mais apropriado para submissão de seus componentes. Desse modo, o usuário pode alterar o arquivo de configuração para apontar para um diretório que seja de fato um repositório

web.

Foi feita uma alteração na funcionalidade de empacotamento de componentes. Incluiu-se a opção de se empacotar todos os componentes, separadamente, com apenas um único comando. Originalmente, podia-se empacotar apenas um componente por vez, ou então todos os componentes em um único pacote. Agora, pode-se empacotar todos os componentes em seus respectivos pacotes individuais de uma vez só. Isso foi implementado utilizando-se chamadas recursivas na função empacotadora à medida que se caminha na estrutura padrão do ACPM (platforms, processors, ip, is, sw e wrappers), passando como argumento o tipo do componente e o próprio componente encontrado. Contudo, manteve-se a possibilidade de gerar um único arquivo contendo todos os componentes empacotados juntos.

A última modificação feita foi no Makefile do ARP. Originalmente, o Makefile era um único arquivo. No ACPM, o Makefile divide-se em 3 arquivos: Makefile, Makefile.rules e Makefile.conf. No Makefile está a plataforma que se deseja executar. No Makefile.rules estão as regras de compilação e execução comuns a todos os componentes e plataformas gerenciados pelo ACPM. No Makefile.conf estão os paths para o SystemC, ArchC e TLM, além da variável de ambiente que define o sistema operacional do usuário. Essa ideia de separação do Makefile do ARP em 3 foi expandida no ESLBench, visto que o ESLBench gera dinamicamente o arquivo Makefile para cada uma de suas plataformas a serem testadas. Essa separação do Makefile facilita a edição do usuário, visto que ele só precisa alterar um único arquivo simples para mudar a plataforma que ele deseja executar ou compilar no momento. Isso também facilita a edição dos paths para as ferramentas de compilação (SystemC, ArchC, TLM), visto que isso normalmente varia de máquina para máquina. O único arquivo que o usuário não precisa editar é o Makefile.rules, visto que este arquivo contém as regras de compilação e execução das plataformas segundo o que foi definido no ARP. Esses 3 arquivos de Makefile funcionam de modo que o Makefile principal dá um include nos outros 2 arquivos, atuando perante o make(1) como se fosse um único grande arquivo de Makefile. Essencialmente, essa foi uma decisão de projeto visando melhorar a experiência do usuário com o ACPM.

Essas foram todas as modificações feitas no código original do gerenciador do ARP até agora. Algumas funcionalidades novas estão previstas, mas ainda estão em desenvolvimento. O objetivo deste artigo é expor as modificações e funcionalidades implementadas até agora.

5. Resultados

O principal resultado, até agora, é a primeira versão do ACPM, que se encontra disponível para download no site do ArchC. Um paper relacionado ao ESLBench foi submetido ao periódico do IEEE “Embedded Systems Letters”.

Além disso, a estrutura básica de diretórios do ACPM é amplamente utilizada no LSC no desenvolvimento e pesquisa de sistemas embarcados. A estrutura também é utilizada na disciplina de Laboratório de Projetos de Sistemas Computacionais, oferecida no curso de Engenharia de Computação da Unicamp. As funcionalidades de obtenção de componentes do ACPM começarão a ser mais utilizadas assim que forem coletados mais componentes para serem disponibilizados no repositório padrão da ferramenta.

O ESLBench utiliza amplamente o ACPM, tanto na estrutura organizacional como

na estruturação do Makefile. A estrutura organizacional é crucial para gerar as diferentes plataformas a serem testadas no benchmark, visto que se todas as plataformas fossem ser mantidas separadamente, haveriam centenas de plataformas distintas. Utilizando-se a estrutura, é possível reduzir o número de componentes distintos para algumas poucas dezenas. Além disso, a separação do Makefile é crucial para o benchmark, visto que este gera automaticamente o Makefile para cada plataforma, de modo a permitir a compilação e execução da mesma. Em suma, é possível considerar o ESLBench o principal resultado prático derivado do ACPM.

6. Conclusões

Gerenciar as diversas versões de um sistema embarcado pode levar inconsistências entre os componentes, gerando diversos erros. Além disso, a distribuição descentralizada de componentes também colabora para essas mesmas inconsistências. A ferramenta ACPM surge para fornecer um ambiente de trabalho organizado e estruturado, além de permitir a fácil obtenção de componentes de uma fonte confiável através da distribuição por repositórios web.

A ferramenta está em constante desenvolvimento e expansão, e serão inclusas funcionalidades de distribuição de ferramentas como o SystemC e ArchC através do ACPM, além de uma interface gráfica para maior facilidade de uso da ferramenta.

O ACPM se encontra para download no site do ArchC, assim como alguns componentes e plataformas de exemplo. O repositório padrão atual está em expansão, e em breve contará com diversos componentes e plataformas, prontos para uso educacional ou científico.

7. Agradecimentos

Este trabalho foi financiado com recursos do CNPq.

References

- Azevedo, Rodolfo and Rigo, Sandro and Bartholomeu, Marcus and Araujo, Guido and Araujo, Cristiano and Barros, Edna (2005). The ArchC Architecture Description Language and Tools. In Azevedo et al, editor, *International Journal of Parallel Programming*, pages 453–484. Springer Netherlands.
- Azevedo, Rodolfo and Albertini, Bruno and Rigo, Sandro (2010). ARP: Um Gerenciador de Pacotes para Sistemas Embarcados com Processadores Modelados em ArchC. In Azevedo et al, editor, *Workshop de Sistemas Embarcados - WSE*, SBC.
- Rigo, S. and Araujo, G. and Bartholomeu, M. and Azevedo, R. (2004). ArchC: a SystemC-based architecture description language. In Rigo et al, editor, *16th Symposium on Computer Architecture and High Performance Computing, 2004*, pages 66–73. SBAC-PAD.