ISSN: 2237-0234

MIGRAÇÃO DA ARQUITETURA DOS SISTEMAS DE AUTOMAÇÃO DA LTQ DA APERAM DE OPENVMS PARA LINUX*

Gláucio Barros Barcelos[1]
Felipe Grativol Lima[2]
Fernando José Mendonça[3]

Resumo
Os sistemas de automação Nível 2 da LTQ da Aperam estão em operação a mais de uma década. Esses sistemas foram desenvolvidos sobre plataforma de hardware Alpha Server DS20E e sistema operacional OpenVMS 7.2. Apesar da confiabilidade da plataforma utilizada, atualmente está descontinuada pelo fabricante, bem como o serviço de suporte ao sistema operacional utilizado que também foi encerrado. Com a evolução da tecnologia Intel, apresentando baixo custo de hardware, suporte, manutenção e desempenho, foi tomada a decisão de migração dos sistemas de automação de Nível 2 da LTQ para a arquitetura de hardware Intel e sistema operacional Linux. Esta migração teve como objetivo a eliminação da obsolescência a que os sistemas estavam submetidos. A migração envolveu uma estratégia que minimiza riscos e reduz modificações necessárias nos sistemas de automação envolvidos. Dessa forma, bibliotecas de funções com funcionalidades semelhantes as disponíveis no sistema OpenVMS foram desenvolvidas. Este artigo descreve todo esforço envolvido na migração dos sistemas e os resultados obtidos.
Palavras-chave: Migração; Linux; OpenVMS; Obsolescência.

APERAM HSM AUTOMATION SYSTEM ARCHITECTURE MIGRATION FROM OPENVMS TO LINUX

Abstract
The current Aperam HSM Level 2 Automation systems have been operating for more than a decade. These systems were developed based on Alpha Server DS20E hardware platform and operating system OpenVMS 7.2. Despite the reliability of the used platform, the current hardware is discontinued by the manufacturer and the operating system reached end of support. With the Intel technology evolution, with a low-cost hardware, support, maintenance and performance, a decision was made to migrate the Level 2 automation systems for Intel architecture and Linux. The migration purpose was elimination of obsolescence that the systems were submitted. The migration strategy involved minimizes risk and reduces necessary modifications in the involved automation systems. Therefore, a set of tools and libraries with similar features available on the OpenVMS were developed. This article describes the effort involved in the systems migration and results.
Keywords: Migration; Linux; OpenVMS; Obsolescence.

[1] *Analista de Automação, Gerência de Automação e Instrumentação, Aperam América do Sul S.A, Timóteo, Minas Gerais, Brasil.*
[2] *Analista de Automação, Gerência de Automação e Instrumentação, Aperam América do Sul S.A, Timóteo, Minas Gerais, Brasil.*
[3] *Engenheiro Eletricista, MENDS Projeto Consultoria Ltda, Timóteo, Minas Gerais, Brasil.*

ISSN: 2237-0234

1 INTRODUÇÃO

Atualmente a laminação de tiras a quente (LTQ), possui um sistema de automação que é composto por dois sistemas de Nível 2 que estão em operação a mais de uma década. Um dos sistemas é responsável pelo gerenciamento e controle do aquecimento das placas em dois fornos de reaquecimento e o outro é responsável pelo gerenciamento e controle do processo de conformação mecânica no laminador de desbate (Rougher) e no laminador de acabamento (Steckel). Estes sistemas de Nível 2 têm as seguintes funções:

- Integração dos sistemas de automação e sistemas coorporativos;
- Interface com o operador;
- Setup dos equipamentos;
- Processamento de modelos matemáticos.

A arquitetura de hardware e software utilizada por estes sistemas de Nível 2 é baseada em tecnologia RISC (Reduced Instruction Set Computer) que utilizam processadores da família Alpha-AXP rodando o sistema operacional OpenVMS 7.2 (neste artigo denominado apenas como VMS).

Os computadores, servidores Alpha Server DS20E, são de fabricação da Hewlett Packard (HP) e foram descontinuados pelo fabricante em 2007. Portanto, não se encontra mais disponíveis peças e componentes novos no mercado. Além disso, o serviço de suporte do fabricante foi encerrado 2012.

Os computadores empregados nesta arquitetura original são equipamentos de alta confiabilidade com um baixo índice ou ocorrência de defeitos. Esta situação conduz a uma falsa tranquilidade ou conforto uma vez que não existe histórico de parada que justifique a troca dos equipamentos, segundo este critério. Contudo, a cada vez que o equipamento é reparado uma peça de segunda mão ou recondicionada é introduzida no equipamento o que reduz o período médio entre falhas (MTBF - Mean Time Between Failures) e a sua confiabilidade.

A tecnologia Intel evoluiu substancialmente nos últimos anos, apresentando baixo custo de hardware, suporte, manutenção e desempenho compatível a de computadores de tecnologia RISC, conforme pode ser visualizado na Figura 1(a). O crescimento da popularidade e robustez do sistema operacional Linux, Figura 1(b), o caso de sucesso de migração do sistema de Nível 2 do laminador de bobinas número 4 da plataforma HP-UX para Linux [1], foi tomada a decisão de migração dos sistemas de automação de Nível 2 da LTQ para a arquitetura de hardware Intel e sistema operacional Linux. Como todos os códigos fonte dos sistemas estão disponíveis e codificados em linguagem C e Fortran é viável a migração do sistema.

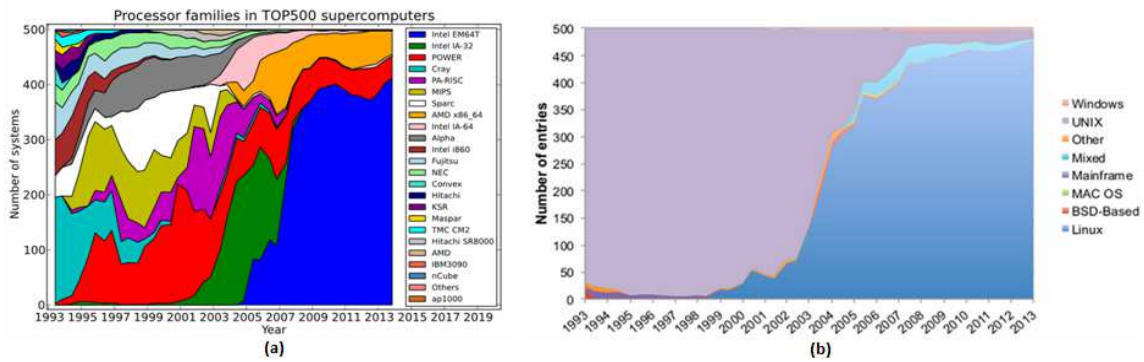

Figura 1. Supercomputadores na lista TOP500 nos últimos 20 anos**: (a)** Família de microprocessadores. **(b)** Sistemas operacionais.

2 DEFINIÇÕES GERAIS

Na arquitetura original, o software do sistema de Nível 2 do forno é escrito basicamente utilizando linguagem Fortran 77 (F77) com o DEC Fortran Extensions e linguagem C com o DEC C. O sistema de Nível 2 dos laminadores é também desenvolvido utilizando linguagem C também com o DEC C.
Em ambos, grandes quantidades de chamadas específicas do sistema operacional OpenVMS são realizadas.
O esquema de eventos assíncronos dos processos dos sistemas está incorporado dentro do sistema de eventos do VMS e a comunicação entre processos é baseada na interface de programação de aplicativos (API) do sistema operacional OpenVMS. Cada sistema, por sua vez, utiliza um conjunto de funções proprietária que foi desenvolvida utilizando funcionalidades do sistema operacional. Dessa forma, todo o sistema está envolvido em um complexo ambiente que utiliza o VMS e um conjunto de APIs que fornecem um conjunto de funcionalidades para aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.
Os usuários dos sistemas de Nível 2, não acessam a interface do VMS diretamente. Todo acesso aos sistemas é realizado através do sistema de supervisão. O sistema de automação dos fornos possui sistema de supervisão desenvolvido utilizando FactoryLink 7.1 e a troca de informação com o sistema de Nível 2 é realizada através de leitura e escrita de arquivos binários. Já o sistema de supervisão dos laminadores foi desenvolvido utilizando Visual Basic 6.0 e a troca de informações com o sistema de Nível 2 é realizada através de conexão TCP/IP e UDP/IP.

2.1 Estratégia de Migração

A estratégia de migração foi formulada com o objetivo de minimizar riscos e reduzir ao máximo qualquer tipo de modificação necessária nos sistema de supervisão, de forma a tornar transparente para os operadores à migração dos softwares de Nível 2. Assim, toda a forma de comunicação e troca de informações entre os sistemas de supervisão e os sistemas de Nível 2 deveriam ser mantidos.
Toda forma de navegação e estrutura de diretórios das tarefas dos sistemas deveria ser respeitada na nova arquitetura do sistema, bem como a navegação entre os diretórios.
Com o objetivo de manter toda a estrutura de código fonte dos sistemas e realizar o mínimo necessário de modificações e adaptações, uma biblioteca de funções da API do VMS deveria ser desenvolvida em Linux. Basicamente, esta biblioteca deveria conter todas as funções específicas do VMS utilizadas pelos sistemas de Nível 2 utilizando APIs e recursos disponíveis no Linux. Dessa forma, a interferência no código fonte dos sistemas seria reduzida, mantendo-se as mesmas chamada de funções de sistema existentes arquitetura anterior.

2.1.1 Compilador Intel Fortran para Linux

O compilador DEC Fortran suporta todos os padrões ANSI FORTRAN-77 e possui um conjunto de extensões/características que tem o objetivo fazer uso de características especiais do hardware e sistema operacional, e também aumentar as possibilidades para programadores Fortran. Um exemplo é o tipo STRUCTURE para criar estruturas de dados como o tipo STRUCT na linguagem C e também diretivas

do compilador como %REF, %LOC e %VAL que possibilita os programadores trabalharem com ponteiros.
O GNU Fortran suporta a maioria das extensões do DEC Fortran. Porém uma extensão fundamental não é suportada que a criação de estruturas. Esta funcionalidade foi padronizada no Fortran 90 com uma sintaxe diferente, sob o nome de TYPE e é possível converter STRUCTURE em TYPE, porém grande quantidade de modificações e adaptações no software seriam necessárias. Dessa forma, testamos com sucesso o compilador Intel Fortran para Linux, que possui a característica para criação de estruturas. Ainda assim, alguns recursos não são suportados como, por exemplo, a diretiva de compilador %DESCR, devendo o código ser adequado a padrão de construção do Fortran 90.

2.1.2 Compilador GNU Compiler Collection - GCC

O GNU Compiler Collection (GCC) possui interface para diversas linguagens, que inclui linguagem C. É distribuído pela Free Software Foundation (FSF) sob os termos da GNU GPL e está disponível para sistemas operacionais Linux [2]. O compilador C fornecido pelo GCC foi utilizado para compilação e link-edição dos programas em C dos sistemas de Nível 2 dos fornos e dos laminadores bem como no desenvolvimento da biblioteca de funções do VMS.

3 DESENVOLVIMENTO

3.1 Biblioteca de Funções VMS

Como mencionado anteriormente, para minimizar a necessidade de modificações nos softwares dos sistemas de Nível 2 uma biblioteca de funções foi desenvolvida. Estas funções possuem os mesmos nomes, número e tipos de parâmetros das funções que são disponibilizadas pela API do VMS e utilizadas nas chamadas de sistema que o software original em VMS realiza.
Toda a biblioteca foi desenvolvida em linguagem C utilizando os recursos que o sistema operacional Linux disponibiliza. Assim, essa biblioteca se torna uma API que fornece interface para os softwares desenvolvidos em linguagem Fortran e C bem como os estados de retorno das funções da API do VMS. Esta biblioteca não contempla todos os serviços oferecidos pela API do VMS, mas sim as funções mais importantes e utilizadas pelos sistemas de Nível 2.
A seguir são apresentados os aspectos mais importantes no desenvolvimento da biblioteca sem entrar muito em detalhes de como foi construída toda a biblioteca.

3.1.1 Eventos – Event Flag

Uma característica fundamental do sistema operacional OpenVMS é o suporte a construção de programas utilizando um paradigma orientado a eventos [3]. Um sistema em tal paradigma, geralmente consiste basicamente em um laço de repetição de eventos, que espera por eventos e repassa os mesmos para o manipulador de eventos. O método pelo qual a informação sobre o evento é adquirida pelas camadas mais baixas do sistema é irrelevante. As entradas podem ser enfileiradas ou interrupções podem ser gerados para reação do programa.
No VMS, existe um método de sincronização semelhante a um semáforo binário, chamado de event flag. Um event flag tem dois valores possíveis, clear e set e quatro operações básicas podem ser realizadas sobre um event flag:

- Operação de ativação ou sinalização: ativa e define o valor de um evento em set.
- Operação de limpeza: limpa e define o valor de um evento em clear;
- Operação de espera: suspende a execução do programa até a ativação de um ou mais eventos;
- Operação de leitura: verifica o estado atual de evento.

Para reconstruir tal paradigma em Linux a biblioteca de funções desenvolvida utiliza as chamadas de sistema eventfd() do sistema operacional Linux para criação e manipulação de eventos [4].

3.1.2 Temporizadores – Timers

O VMS permite acoplar temporizadores a eventos. Ou seja, um temporizador permite um processo agendar uma notificação futura para si mesmo que será ativada na expiração do tempo especificado. Quando o tempo expira, um evento é ativado e opcionalmente uma rotina pode ser executada.
No Linux, foi utilizada uma API POSIX conhecida com POSIX (interval) timers para criação, ativação e cancelamento de temporizadores [5].

3.1.3 Fila de Mensagens – Mailbox

No VMS um Mailbox é um dispositivo virtual criado para permitir comunicação entre processos. As mensagens escritas em um Mailbox são armazenadas em forma de fila FIFO (First In, First Out). Ou seja, as mensagens vão sendo colocadas e retiradas (ou processadas) por ordem de chegada. Um Mailbox pode ter vários escritores e vários leitores [6]. Porém, geralmente é mais fácil de implementar um único leitor para um Mailbox conforme mostra a Figura 2(a).

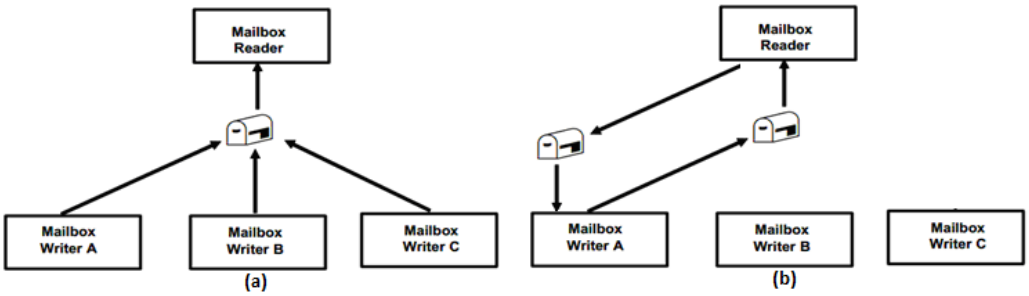

Figura 2. (a) Fila de mensagem com um leitor e vários escritores. **(b)** Comunicação entre fila de mensagens do escritor e leitor

Quando o escritor quer obter mensagens de volta a partir do leitor, a forma mais comum é a criação de um outro Mailbox passando adiante a identificação do Mailbox para o leitor. Dessa forma o leitor pode enviar uma resposta para o Mailbox do escritor como na Figura 2(b).
Como no caso dos temporizadores, o VMS também permite acoplar a chegada de uma mensagem a eventos. Dessa forma, um processo pode solicitar notificação assíncrona da chegada de uma mensagem em uma fila anteriormente vazia.
No Linux, para desenvolvimento das funcionalidades de um Mailbox foi utilizada a API POSIX Message Queues para permitir a comunicação entre processo na forma de troca de mensagens [7].

ISSN: 2237-0234

3.1.4 Memória Compartilhada – Global Sections

O serviço de gerenciamento de memória do VMS permite que dois ou mais processo possam se comunicar através de memória compartilhada chamadas de Global Sections. Através de Global Sections, um processo pode mapear em seu espaço de memória virtual a região de um arquivo em disco ou região de endereços físicos da memória.

No Linux foram utilizadas chamadas de sistemas mmap() para criar mapeamentos de memória no espaço de endereçamento virtual do processo [8]. Esse processo de mapeamento foi dividido em duas categorias:

- Mapeamento de arquivo: mapeia uma região de um arquivo para a memória virtual do processo de chamada. Quando o último processo tiver terminado de trabalhar com o arquivo, os dados são salvos no arquivo de origem no disco;
- Mapeamento anônimo: não tem um arquivo correspondente. Quando o último processo tiver terminado de trabalhar com o arquivo, os dados são perdidos.

3.1.5 Arquivos Indexados - ISAM

No VMS o serviço de gerenciamento de registros (RMS) fornece uma variedade de procedimentos que auxiliam os programas de aplicação no processamento e gerenciamento de arquivos e seus conteúdos.

O RMS suporta organizações de arquivos de forma seqüencial, relativa e indexada. Onde o formato do registro pode ser fixo ou variável para cada tipo de organização de arquivos.

Os modos de acesso ao registro no RMS permitem acessar registros sequencialmente, diretamente pelo valor de chave ou diretamente pelo número relativo do registro.

No Linux, não existe um API nativa que fornece este tipo de funcionalidade. Dessa forma uma biblioteca de funções com o código fonte aberto chamada de PBL (The Program Base Library) registrada sobre os termos da licença GPL (GNU General Public License) foi utilizada [9].

A biblioteca PBL inclui diversos módulos, dentre eles um que implementa uma API C ISAM de código fonte aberto. ISAM é o acrônimo para Indexed Sequential Access Method (ISAM) sendo um método para a indexação de dados para recuperação rápida. Dados ISAM estão organizado em registros que são armazenados em arquivos de dados. Arquivos de índice separados são usados para armazenar valores de chave que identificam cada registro, junto com ponteiros que servem para localizar o registro correspondente no arquivo de dados. O índice faz com que seja possível recuperar os registros individuais de forma rápida sem ter que procurá-los em um conjunto de dados inteiro.

No sistema de Nível 2 dos laminadores, existe uma API desenvolvida pela VAI Siemens que fornece funcionalidades para criação e manipulação de dados em arquivos indexados utilizando RMS do VMS. Esta API foi modificada, substituindo as chamadas de sistema do RMS pelas chamadas de sistemas fornecidas pela PBL e incorporada ao sistema de Nível 2 dos fornos. Dessa forma todo o código fonte dos sistemas de Nível 2 não sofreu grande modificações para tratamento de arquivos indexados.

3.2 Driver de Comunicação GE Fanuc

Para comunicação com os PLCs GE Fanuc Series 90-70, os dois sistemas utilizam uma biblioteca desenvolvida pela GE Fanuc chamada de Host Communication Toolkit (HCT), onde o código fonte dessa biblioteca não está disponível. Dessa forma, uma nova biblioteca foi desenvolvida e simplesmente as função principais e utilizadas pelos sistemas de Nível 2 foram implementadas.
Basicamente foram desenvolvidas as chamadas de sistema para conexão e desconexão, chamadas para leitura e escrita em um bloco contíguo de dados a partir de qualquer tipo de memória do PLC, bem como leitura e escrita de dados dentro de um segmento local de dados de um sub-bloco denominado memória local (%L).
Para suportar o acesso à memória do PLC a GE Fanuc definiu um protocolo proprietário denominado Service Request Transfer Service (SRTP) que utiliza conexões TCP/IP padrão para transferir as informações entre o PLC e o computador através de troca de mensagens. Essas mensagens não são documentadas no HCT, mas as mensagens SRTP são semelhantes às mensagens SNP (Series Ninety Protocol) que é um protocolo serial utilizado para comunicação [10]. Nas mensagens do SRTP apenas a parte de dados referente ao Mailbox nas mensagens SNP são utilizadas em conexão TCP. Dessa forma uma biblioteca de funções que implementa o protocolo SRTP foi criada e incorporada aos sistemas de Nível 2.

3.3 Migração dos Sistemas de Nível 2

Após a definição da estratégia de migração, foi elaborado um cronograma de trabalho com previsão de 6 meses para cada sistema de Nível 2. Em seguida foi montada uma equipe de trabalho composta por especialistas em sistema operacional (OpenVMS e Linux), redes e linguagem de programação C e Fortran e conhecimento do sistema Nível 1 (PLCs GE Fanuc e ABB). O cronograma proposto e realizado contemplou algumas fases que serão detalhadas a seguir.

3.3.1 Instalação e montagem do ambiente de desenvolvimento

Nesta fase foi realizada a instalação e configuração do sistema operacional Linux bem como a configuração do ambiente de desenvolvimento. Foram criadas, por exemplo, as contas de usuários, a estrutura de diretórios de arquivos, procedimentos para compilação e link-edição dos arquivos fontes e cópia dos arquivos do código fonte dos sistemas de Nível 2 para o ambiente em Linux.
Foi realizada também a instalação do software do Banco de Dados Oracle e também o Pré-compilador Oracle (Pro*C), utilizado para realizar acesso ao banco de dados a partir de programas desenvolvidos em linguagem C.

3.3.2 Compilação do sistema

Nesta fase foram realizados diversos testes, além de trabalho de pesquisa que permitisse o entendimento detalhado do código fonte do sistema para que pudessem ser definidas as adaptações e correções necessárias no código fonte do sistema.
Após o estudo detalhado do código fonte, foi desenvolvida a biblioteca VMS com as funcionalidades apresentadas anteriormente, a biblioteca com o driver de comunicação com PLCs GE Fanuc e outras bibliotecas e funções que não foram mencionadas anteriormente.

Em seguida todos os arquivos de código fonte foram compilados incluindo os módulos relativos ao modelo matemático, sendo corrigidos os erros oriundos do processo de migração motivado pela mudança de plataforma.

3.3.3 Testes e comissionamento

Após o desenvolvimento e compilação de todo sistema, foram realizados testes para verificação do funcionamento de cada modulo do sistema de Nível 2. Foram realizados os testes de comunicação entre os sistemas de Nível 2 com as estações clientes e os sistemas de Nível 1 e 3.

Para o sistema de Nível 2 dos fornos, foi montada uma plataforma de teste utilizando um PLC GE Fanuc reserva. Esta plataforma foi utilizada para testar as funcionalidades do driver de comunicação GE desenvolvido (HCT) realizando operações de leitura e escrita no PLC. Após a garantia de funcionamento da biblioteca o sistema de Nível 2 migrado foi colocado operando em paralelo ao sistema original para teste de todo o sistema com o objetivo depurar eventuais erros ainda remanescentes e análise dos resultados.

Diferente do PLCs GE Fanuc do sistema de controle dos fornos, os PLCs da ABB (Advant Controller) e a maioria dos equipamentos de controle tecnológico (medidor de espessura, largura e perfil, entre outros) do sistema de controle dos laminadores não foram projetados ou programados para gerenciar e manipular conexões de mais um sistema de Nível 2. Como a maioria destes equipamentos possui código fonte fechado e a CPU dos PLCs ABB não possuíam memória suficiente para desenvolvimento de funcionalidade, foi desenvolvido um analisador de rede (Sniffer) capaz de interceptar os pacotes de dados que trafegam na rede.

Este analisador de rede captura os pacotes, decodifica e analisa o seu conteúdo, descartando pacotes não pertinentes a aplicação e reenviando as informações importantes para o sistema de Nível 2 em ambiente de desenvolvimento (Linux).

Com a plataforma montada, foi possível realizar testes de funcionamento integrado de todo o sistema de Nível 2, bem como a comunicação entre os dois sistemas de Nível 2 e também comparar os resultados do modelo matemático com os resultados do sistema original.

4 RESULTADOS E DISCUSSÃO

4.1. Domínio do Sistema de Nível 2

Após a execução do trabalho a equipe envolvida passou a ter domínio total do software (código-fonte) dos dois sistemas de Nível 2, sendo obtido compreensão do funcionamento de todas as funções e rotinas implementadas.

4.2. Ambiente de Desenvolvimento

A arquitetura original do sistema de Nível 2 dos fornos, não possuía ambiente de desenvolvimento para realização de testes em modificações ou desenvolvimento de novas funcionalidades. As modificações eram realizadas e testadas no sistema em produção, aumentando o risco de interrupção do fluxo de produção da planta. Com a migração do sistema, foi possível criar um ambiente de desenvolvimento para o sistema que executa o mesmo software, porém é bloqueada a funcionalidade de escrita e sendo habilitados apenas a leitura de informações. O sistema em ambiente de desenvolvimento recebe as mesmas informações que o ambiente de produção,

possibilitando analisar os resultados das modificações e comparar com os resultados do sistema em produção antes de implantar as novas funcionalidades.
O sistema de Nível 2 dos laminadores possuía ambiente desenvolvimento, porém não recebia informações reais do processo. Os testes eram limitados à simulação de alguns eventos. Com o desenvolvimento da ferramenta de Sniffer, foi possível enviar as informações do processo para a aplicação em ambiente de desenvolvimento permitindo também analisar os resultados das modificações antes de serem implantadas na produção.
O ambiente de desenvolvimento para os sistemas é importante, pois diminui a possibilidade de erros e necessidade de refazer o trabalho em caso de erros de programação, aumentando a produtividade e a capacidade de manutenção no software.

4.3 Lateralidade

O sistema de Nível 2 dos fornos foi desenvolvido pela Stein Heurtey. E este mesmo sistema foi utilizado em fornos de reaquecimento de placas de outras empresas. A solução de migração adotada aplica-se totalmente a estes sistemas e outros sistemas que utilizam VMS, sendo necessário apenas algumas adaptações e desenvolvimento de alguma funcionalidade específica de cada planta e/ou sistema.

4.4 Custo

A migração dos sistemas para uma nova plataforma, mantendo a utilização do OpenVMS também era possível com a utilização de servidores HP Integrity com processadores Itanium. Com a migração para essa plataforma de hardware, havia a necessidade de migração da versão do sistema operacional OpenVMS 7.2 para a versão 8.3 e com a mudança da versão do sistema operacional seriam necessárias adaptações no software da mesma forma, pois nem todas as chamadas de sistema do OpenVMS 7.2 são compatíveis com a versão 8.3.
Dessa forma, a aquisição da infraestrutura de hardware, licenças de software e horas de trabalho para manter o OpenVMS era superior à aquisição dos mesmos equipamentos da arquitetura Intel. Mesmo com uma desvantagem inicial a previsão de trabalho de migração de software mais extensa com um possível número maior de correções a serem efetuadas, a migração do sistema para plataforma Linux se torna vantajosa.

4.5 Atualização Tecnológica

A atualização tecnológica do sistema elimina vulnerabilidade operacional devido ao risco de parada no equipamento em caso de pane nos servidores uma vez que a Aperam não possui contrato de manutenção para o hardware e também elimina a obsolescência a que o sistema estava submetido.
Foram observados ganhos relacionados ao tempo gasto de compilação nos dois sistemas. Atualmente, o tempo gasto para compilar os sistemas de Nível 2 gira em torno de 15 a 30 minutos e na nova plataforma este tempo caiu para um valor em torno de 3 minutos.
Durante a migração puderam também ser detectados e corrigidos alguns erros existentes no software nativo, que poderiam, em determinadas condições, levarem a cálculos incorretos do modelo matemático ou falhas no sistema.

Outro benefício da atualização tecnológica é a utilização de interface gráfica para navegação no sistema e facilidade e tempo gasto para execução de cópias de segurança (backup) dos diretórios do aplicativo bem como imagem dos discos físicos.

5 CONCLUSÃO

O estado atual do projeto demonstra a viabilidade técnica de migração respeitando estratégia de mínima modificação do código fonte e interoperabilidade com os sistemas de supervisão dos fornos de reaquecimento e laminadores.
As bibliotecas e ferramentas desenvolvidas permitiram construir uma base sólida para a migração que facilitaram e permitiram que os testes fossem realizados de rápida e eficientes sem grandes modificações nos softwares.
E por fim, a atualização tecnológica reduz o custo com a manutenção do hardware e permite evolução tecnológica sempre que necessário, uma vez que o custo do hardware é menor e o sistema operacional Linux está constante evolução.

REFERÊNCIAS

1 Mendonça, Fernando José; Jesus, Geraldo Tavares de. Uso de LINUX em ambiente industrial, o caso LB4 da Acesita. Timóteo (MG) Acesita S.A; 2005. RT43-001/2005.
2 Projeto GNU [homepage na internet]. The GNU Compiler Collection [acesso em 10 mar 2015]. Disponível em: https://gcc.gnu.org/
3 R. Huhmann, G.Fröhlich, S. Jülicher, V.RW Schaa, “GSI Operating Software Migration Openvms to Linux”, Proceedings of PCaPAC08, Ljubljana, Slovenia, MOX02, p. 4.
4 EVENTFD(2) Linux Programmer's Manual [homepage na internet]. The Linux man-pages project [acesso em 10 mar 2015]. Disponível em: http://goo.gl/Br1YWo
5 TIMER_CREATE(2) Linux Programmer's Manual [homepage na internet]. The Linux man-pages project [acesso em 10 mar 2015]. Disponível em: http://goo.gl/1grU3Z
6 OpenVMS Technical Journal V9 [homepage na internet]. OpenVMS Mailboxes: Concepts, Implementation, and Troubleshooting [acesso em 10 mar 2015]. Disponível em: http://h71000.www7.hp.com/openvms/journal/v9/mailboxes.pdf
7 MQ_OVERVIEW(2) Linux Programmer's Manual [homepage na internet]. The Linux man-pages project [acesso em 10 mar 2015]. Disponível em: http://goo.gl/20uzqy
8 Michael Kerrisk. *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*. No Starch Press, 1 edition, 10 2010.
9 PBL - The Program Base Library [homepage na internet]. Peter Graf's Free GPL Open Source Software [acesso em 10 mar 2015]. Disponível em: http://www.mission-base.com/peter/source/
10 Series 90 PLC SNP Communications User's Manual [homepage na internet]. Series 90* PLC SNP Communications [acesso em 10 mar 2015]. Disponível em: http://goo.gl/OeYPrx