

SPM-Linux/FX2

R.V. Ribas

20 de julho de 2013

1 Introdução

Esta nova versão do SPM tem pequenas modificações para utilizar o novo sistema chamado CAMAC-USB, que faz a interface da instrumentação CAMAC com o barramento USB de microcomputadores. Nos módulos de aquisição (spm, scan, etc.) não há modificações perceptíveis ao usuário. A única alteração visível consiste no novo modo de se programar o “Event Handler” que agora é um programa escrito em linguagem c e executado no microcontrolador Cypress FX2. Entretanto, o formato do novo programa é muito similar ao anteriormente utilizado. Veja abaixo instruções para criar e compilar o programa de tratamento de eventos.

2 Introdução Original

O sistema de aquisição de dados (SPM-PC), foi modificado para rodar sob o sistema operacional Linux, para o mesmo hardware (PC-Pentium). A utilização dos programas de aquisição não sofreu alterações significativas (i.é.: os comandos utilizados são exatamente os mesmos). As principais vantagens são a maior estabilidade deste sistema em relação ao Win95, maior segurança na manutenção dos arquivos dos grupos e a possibilidade de se controlar remotamente a aquisição (ssh), bem como de visualização e análise dos dados em tempo real (**dammm**), de qualquer X-terminal (vários **damms** podem simultaneamente acessar os histogramas de aquisição). Além disso, com o aumento da memória *ram* do micro de aquisição (spmnt), foi aumentada a capacidade de histogramas durante a aquisição (agora até 64 Mbytes). No sistema SPM-Linux, o processamento é distribuído em vários processos, de maneira similar ao que acontecia na primeira versão do SPM, sob o Vax-VMS. No presente caso, são 4 processos, comandados por um pequeno programa, que recebe os comandos do usuário e controla a distribuição de comandos para os outros 3 módulos. O módulo de aquisição (**spm-linux**) é responsável pelo controle do camac, gravação de dados na fita (ou disco, dependendo da escolha do modo fila) e colocar os buffers de dados para outro processo, o **scanaq-linux**, que faz a análise em linha. Em princípio, nem todos os buffers adquiridos são processados pelo **scanaq-linux**. A prioridade é dados para a fita e se o **scanaq-linux** tiver uma carga muito grande, uma fração dos buffers não será processada. Esta limitação deve entretanto só acontecer em casos muito especiais, com taxas muito altas de contagens, e

histogramação complexa. O **scanaq-linux** constrói os histogramas num espaço de memória *ram* que pode ser compartilhado com outros processos na mesma *cpu* (*shared memory*). O terceiro processo utilizado no sistema é o **damm**. Este processo é entretanto independente dos outros, podendo ser cancelado ou iniciado a qualquer momento. Vários processos **damm** podem estar ativos simultaneamente, permitindo que os dados de aquisição sejam visualizados em vários terminais.

3 Escrevendo o programa de tratamento de eventos

- Obtenha um modelo para o programa de tratamento de eventos, p.ex. copiando `/usr/local/spm-fx2/fx2/ev-entemplate.c` para `meuevento.c`
- Edite `meuevento.c` para tratar seus eventos, de modo analogo ao que se faria com o `meuevento.evs`
- compile o o programa: `adacfx2 meuevento` para produzir o módulo executável (no microcontrolador do FX2) `meuevento.ihx`
- inicie o `spm-fx2` como usual e carregue o programa `meuevento.ihx` com o comando `>ldac meuevento` ou `ldav meuevento`

3.1 Exemplo de programa

```
unsigned char NN1, NN2, SEC, NP;
#define FX2_ISA //Select the DSP_ISA-USB interface
//#define CAEN_ISA //Select the 4-module Box
//#define FX2_CAM128

#define NBFIF 460 // Event Fifo size. Max 480 (bytes)

void init_camac(void) {
    unsigned char A,i;
    Z();
    NN1=16;
    NN2=17;
    naf(NN1,0,9); //A=0, F=9 clear module NN1
    naf(NN2,0,9); //A=0, F=9 clear module NN2
    CA.W=128; //ULD
    A=0;
    for (i=0;i<8;i++){ naf(NN1,A,17); naf(NN2,A,17); A++; }
    ...
}
```

```

void event(void) {
    pat(NN1,15,0);
    SEC=0;
    NP=0;
    if(PAT.B[0]!=0)
    {
        if((PAT.B[0]&0x01)!=0)
        {
            SEC=1;
            out(0x8001);
            rout(NN1,0,0); //read P1 and send to event buffer (A=0, F=0)
            rout(NN2,0,0); //read P2 and send to event buffer (A=0, F=0)
            NP++;
        }
        if((PAT.B[0]&0x02)!=0)
        {
            if(SEC==0)
            {
                out(0x8003);
            }
            SEC=1;
            rout(NN1,1,0); //read P3 and send to event buffer (A=1, F=0)
            rout(NN2,1,0); //read P4 and send to event buffer (A=1, F=0)
            NP++;
        }
        else SEC=0;
        ...
    }
    nafd(NN1,0,9); //A=0, F=9 - clear module
    nafd(NN2,0,9); //A=0, F=9 - clear module
    flush_event();
}

```

4 Diferenças em relação à versão anterior

São poucas as diferenças para o sistema anterior. Todo o conjunto de comandos dos módulos permanece o mesmo:

- O Arquivo de LOG se chama SPMLOG. A cada novo run do SPM, novos comandos (se for entrado o comando LON) são colocados à partir do final do arquivo (se já existente). É portanto interessante, a cada período de máquina renomear o SPMLOG eventualmente existente. O LOG é somente dos comando para o módulo de aquisição (spm-linux).

- Para carregar um scan personalizado (com usersubs, chamado p.ex. de meuscan, e criado com o comando `scanlnk usersubs meuscan`), usar o comando: `]#spm meuscan`
- O módulo DAMM é carregado independentemente dos outros módulos. Use uma outra janela do KDE para iniciar o DAMM. Isso faz com que eventuais problemas com o DAMM não derrube os outros módulos de aquisição. Lembre-se que comando de abertura do arquivo de display de aquisição (in nome.his) deve vir sempre depois do comando de definição dos histogramas de aquisição (hac nome), feitos em memória, para visualização dos histogramas em tempo real.
- Há um comando `cmdf` para o spm, assim como no DAMM. No comando `cmdf` do spm, o nome do arquivo de comandos deve ser dado incluindo a extensão do nome (`cmdf nome.cmd`).

5 Principais características do sistema:

- A análise em linha é feita com utilização de um processador de linguagem CHIL que constrói os histogramas diretamente na memória (memória compartilhada) O histogramador de aquisição, SCANAQ é capaz de construir até 8MB canais de histogramas, em até 16MB de memória ram. O programa de análise em linha pode ainda ser personalizado, com a utilização de USERSUBs e USERCMP. Com o comando `scanlnk`, pode-se criar um novo processo (meuscan), que será carregado ao invés de scancaq, com o comando `spm meuscan`.
- Qualquer usuário, com privilégios suficientes, pode se logar na estação de aquisição e visualizar os histogramas de aquisição (na memória compartilhada), com a utilização da versão usual de Damm.
- Os programas de apoio, como o compilador CHIL e o assembler do programa do Event-Handler (ADAC) estão também implementados no sistema. Outros programas do Upak, como CCF, KINEQ, STOPX, CARGA e CALIB estão também disponíveis. Lembramos que como é usual no sistema Unix, todos estes comandos foram implementados em letras minúsculas.

Um pequeno manual dos comandos de aquisição, bem como das regras básicas para utilização do novo sistema é apresentado a seguir.

6 Normas Para Utilização do Sistema

Sendo um sistema operacional multi usuário, a utilização do micro computador de aquisição de dados requer um cuidado maior com a desligagem (*shutdown*), sempre tendo em consideração que outros usuários podem estar acessando o computador. Além disso, particularidades

do sistema de arquivo fazem (como em outros sistemas Unix) que o ato de desligar o computador "na chave", sem o procedimento de shutdown, pode trazer sérios danos aos arquivos de dados. Veja mais adiante os procedimentos para desligar o computador. Cada grupo terá uma conta no micro, com áreas de disco no /home/nomedogrupo e no /free/nomedogrupo. No momento não estão habilitadas quotas de disco por grupo. Procure entretanto manter os discos o mais limpo possível. Como base, procure manter o /home e o /free do grupo abaixo de 1Gbytes. LOGOUT: Ao fazer o logout (exit), verifique se não deixou nenhum módulo do sistema de aquisição ativo. Isso pode ser feito com o utilitário Gerenciador de Processos (no KDE, menu K - Utilitários - Gerenciador de Processos. Exiba só os processos do usuário (Ver - Seu Processo) e termine os referentes ao **spm** (em caso de crash, termine todos os processos **spm** (kill) e digite o comando **spm-limpa**). Deixar processos inacabados do **spm**, impede que outros usem o sistema.

7 Desligando o micro computador.

Só desligue o computador, caso haja real necessidade e todos os procedimentos de recuperação do sistema tiverem sido testados. Prefira sempre o auxílio de uma pessoa familiarizada com o Linux, para tomar esta atitude. Lembre-se que outras pessoas podem estar acessando o sistema no momento. Se possível feche todos os processos abertos diretamente na estação, teclando <Ctrl><Alt><BackSpace>. A seguir, tecle <Ctrl><Alt><F1>. Um terminal tty simples (similar a um VT100) deve aparecer na tela. A seguir tecle <Ctrl><Alt>. O sistema deve imediatamente entrar no processo de religagem.

8 Utilizando o Sistema.

- Logando na conta de seu grupo, crie um diretório para o experimento. Há dois discos disponíveis, e há cerca de 8Gbytes em cada um deles (/home/grupo e /free/grupo). Use o comando df para ver o espaço livre nos discos e o comando du para ver o espaço ocupado por seu grupo.

```
|#mkdir experim
```

- Escreva os programas do Event-Handler (experim.evs) e o programa de histogramação (experim.chl). Para editar os arquivos de programa fonte, você pode escolher entre os editores Unix como pico ou emacs.

- Compile os programas:

```
|#adac experim (será criado experim.evo, caso não haja erros)
```

|#chil experim (será criado experim.mil e experim.drr, o tamanho necessário para o arquivo experim.his será indicado (número de blocos de 512 bytes). O arquivo experim.his deverá então ser criado e inicializado com zeros, com o programa ccf.

- Caso sua análise em linha necessite de USERSUBs para operações nos parâmetros, escreva a(s) subrotinas USERSUBs necessárias e eventuais processadores de comando do usuário (USERCMP). Digamos que estas rotinas estão no arquivo chamado experim.f. Compile este arquivo com o compilador fort77, e linke o objeto com o restante das rotinas do

programa de histogramação com o comando `scanlnk`. O novo programa de histogramação, personalizado, será chamado, p. ex. `meuscan`: `]#scanlnk experim meuscan`

- Ligue o crate `camac`. Carregue os programas de aquisição como o comando **spm**. Os comandos para o gerente **spm** podem ser digitados em minúsculas ou maiúsculas. Entretanto, como os nomes de arquivo serão normalmente em minúsculas, recomenda-se utilizar sempre os comandos também em minúsculas: A seguir, uma seqüência típica de comandos, iniciando uma aquisição, com a indicação do módulo que o executa e a ação.

SPM->ldac experim (**spm-linux**: carrega o programa do EH)

SPM->hac experim (**scanaq**: define os histogramas de aquisição)

SPM->run (**spm-linux**: inicia aquisição)

SPM->stop (**spm-linux**: pára aquisição)

SPM->mont (**spm-linux**: monta fita. Atualmente é preferível armazenar os dados em um HD removível de alta capacidade - veja comando **fila**)

SPM->fita lig (ou **tape on**) (**spm-linux**: habilita gravação de modo fila)

SPM->run # Comentários para gravar no header da fita (40 carac.) (**spm-linux**: run)

SPM->fila nome_ (**spm-linux**: raiz para os arquivos de disco para gravação de dados brutos. A cada comando **run**, um novo arquivo `nome_1.fil`, `nome_2.fil`, etc. é aberto. Se o comando **fila** é entrado sem o nome da raiz de arquivo a gravação é desabilitada.

9 Travamento e Recuperação de Falhas dos módulos de aquisição.

Quando o SPM-Linux está rodando, esta informação é registrada no arquivo `/tmp/SPM-Running`. Esta informação é removida quando termina a execução do SPM-Linux. Caso o programa termine de maneira irregular, a informação de que o sistema de aquisição está em uso pode permanecer no arquivo `SPM-Running`, impedindo que o SPM seja recarregado. Nesse caso, use o comando **spm-limpa** para restabelecer a condição inicial. (ATENÇÃO: Só emita este comando estando na sala de controle e estando o seu grupo em tempo de máquina, após verificar cuidadosamente se o sistema de aquisição está realmente fora do ar, e não sendo utilizado por outro grupo!)

Caso algum dos módulos pare de responder, o gerente pode também ser bloqueado, se estiver esperando a resposta p.ex. de fim de comando do módulo. Para reinicializar o sistema:

- Verifique os módulos que ainda estão rodando com o comando: `]#ps`. O sistema mostra o número do processo e o nome de cada programa girando nesta conta. Se o módulo `scanaq-linux` (ou a versão personalizada do `scanaq`) não estiver rodando, você pode reparti-lo. Para isso, na janela em que está o sinal de pronto do `spm-gerente` (`SPM->`), faça `<Ctrl> z` para passar para background. Então tente repartir o processo que abortou, fazendo p.ex.:

|# scanaq-linux & - para repartir o scanaq-linux ou |# ./meuscan & - para repartir o scanaq personalizado, meuscan. Faça o spm-gerente voltar para o foreground com o comando:|#fg

- Caso o **spm-linux** tenha abortado, cancele todos os módulos do sistema (**spm-gerente**, **spm-linux**, **scanaq-linux**) e parta novamente o sistema.

Um outro tipo de problema que pode surgir, corresponde ao bloqueio de memória compartilhada, por outros programas damm que estiver com o arquivo anterior de aquisição aberto, (isto somente se os programas de aquisição foram terminados e carregados novamente). Caso após o comando HAC, tenha uma mensagem de erro do tipo ERROR Getting Shared Memory, procure saber se tem outras pessoas (de teu grupo) acessando os histogramas de aquisição, que devem então fazer o comando clo n, ou clo f, etc. no **damm**. Caso não tenha como avisar estes usuários, cancele os damms abertos, utilizando o utilitário Gerenciador de Processos para obter os PIDs dos processos de nome damm e depois utilizando o menu Matar.

10 Descrição dos Comandos para o módulo spm-linux

HUP: Atualiza no disco (arquivo .HIS definido pelo comando HAC) os histogramas que estão na memória. Não há necessidade de HUP para visualização. O comando HUP só deve ser emitido com a aquisição parada.

ZERO ALL: Zera tanto a memória quanto o arquivo em disco. Se dados estão sendo armazenados (SALVA_FILA ligado), ZERO só pode ser dado em STOP.

ZERO IDN: Zera somente o ID=IDN na memória.

STAT: mostra na tela o status da aquisição: Aquisição parada/andando, nome do programa do EH (se foi carregado), nome do arquivo .HIS para análise em linha, nome do arquivo de modo fila, número de blocos adquiridos, número de possíveis perdas por tempo morto muito grande no PC, número de records (32k) gravados na fita, número de eventos (bons), processados pelo SCANAQ e número de eventos rejeitados pelo SCANAC.

DELS, DELM, DELH: define intervalo de tempo (em segundos, minutos e horas respectivamente) de duração da aquisição após o próximo comando RUN. Note que internamente o intervalo é sempre em segundos. As três definições são somente para maior conforto do usuário. Apenas a ÚLTIMA definição é válida (para fazer 1 min. e trinta segundos faça DELS 90 e não DELS 30 e DELM 1 !!!). Ao se esgotar o intervalo de tempo, a aquisição entra em STOP, e um sinal sonoro é emitido para aviso.

CONT: Ao se dar um comando RUN, o número de blocos adquiridos e o número de eventos são zerados. Caso durante o RUN tenha sido dado um PAUS e quer-se continuar sem zerar a estatística, emita o comando CONT ao invés de RUN.

LON: Log ON - Este é um comando original do DAMM e outros programas do VAXPAK. O arquivo SPMLOG é criado (append é feito caso SPMLOG já exista no diretório de

aquisição) e todos os comandos enviados ao programa e as respectivas mensagens de resposta são copiadas para este arquivo, com dia e hora da execução.

MONT: Monta fita na unidade.

DISM: Desmonta (unload) fita.

BOT: Vai até o fim da fita (2EOF) posicionando para continuar a gravar numa fita que já contem dados.

FILA nomefil : O arquivo nomefil01.fil para armazenagem de modo fila em disco será aberto ao próximo comando RUN. Só deve ser usado em experimentos nos quais a quantidade de dados seja relativamente pequena ($< \sim 1\text{GByte}$ por período). O apêndice numérico do nome (01 acima) é incrementado a cada novo comando RUN. O Comando FILA sem o parâmetro nomefil, fecha o nomefil existente e desabilita o modo fila.

FITA LIG: Modo Fila ligado e dados começam a ser registrados em fita.

FITA DES: Modo Fila desligado. Fita continua montada e posicionada.

RUN NR COMENTÁRIOS: Inicia a aquisição. Se a fita estiver ligada, o programa escreve o header do formato L002 na fita (record de 256 bytes, contendo o número do run NR e o comentário. Isto é importante para posteriormente localizar dados na fita, por exemplo com o comando FIND do SCAN. Se um novo número de RUN não é dado, o valor anterior é incrementado.

BIP N - Alarme de ausência de eventos: Se a taxa de eventos ficar menor que 1 buffer de dados a cada N segundos, um bip é soado. BIP sem parâmetro desliga o alarme.

- No início de cada HIS de aquisição (primeiros $2 \times \text{NPAR}$ canais), é feito automaticamente um histograma com a estatística de cada parâmetro definido na aquisição. Por isso, ao escrever um programa CHIL, defina sempre como primeiro histograma a ser construído (primeiro definido) um para esta função. (Caso contrario, os primeiros N canais do primeiro histograma vão conter esta estatística):

H(1) L(128) G(1 127,127)