IST - 1995/1996

Sistemas Distribuídos para Ambientes Industriais

Trabalho de Laboratório – E286

Controlo de uma Célula de Fabrico

1 Introdução

A competitividade crescente observada globalmente, nos mercados internacionais, tem vindo a colocar exigências cada vez mais fortes na modernização das empresas industriais. Uma das componentes desta modernização é a automatização do processo produtivo.

Uma fábrica flexível será então organizada em torno de células de fabrico, com funções mais ou menos específicas, entre as quais são transferidas matérias primas e/ou produtos semi-fabricados.

Ao nível da célula de fabrico, e falando estritamente em termos computacionais, é vulgar encontrar sistemas compactos, normalmente constituídos por uma base de processador inteligente complementada com elementos de *hardware* e *software* que lhe conferem características funcionais acrescidas: capacidade de entradas/saídas, interfaces com redes de comunicação, capacidade de execução de tarefas em multiprocessamento.

Este trabalho propõe a aplicação dum *sistema embebido* no controlo de uma pequena célula fabril, e destina-se a solidificar, pelo contacto experimental, noções de sincronização e programação concorrente. Para tal será empregue o E286, um executivo de tempo-real, muito semelhante ao iRMX, o qual, sendo lançado por uma aplicação MS-DOS, é executado depois em modo nativo num PC. Estes dois componentes constituem a infra-estrutura computacional que será utilizada no controlo do acesso a um armazém automatizado, efectuado por parte de veículos filoguiados, que suportam a movimentação automática de produtos e matérias primas, numa fábrica flexível (Figura 1).

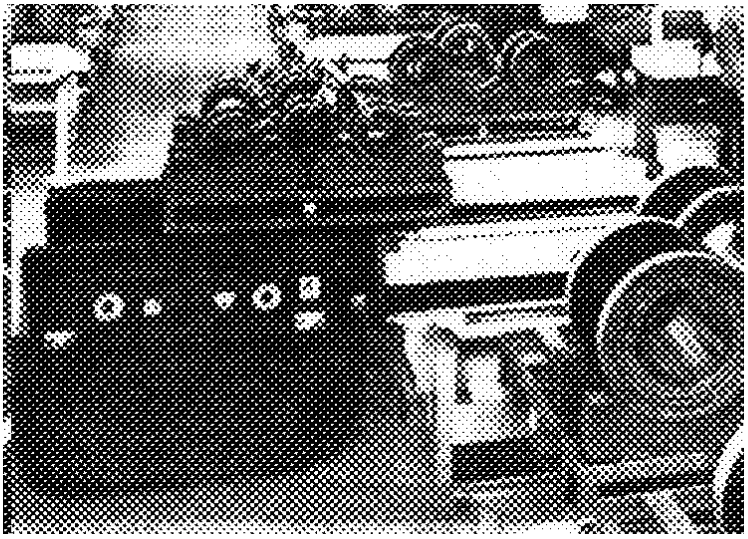

Figura 1: Exemplo dum filoguiado em plena operação.
(EFACEC - Fábrica de Transformadores de Distribuição)

A aplicação que se pretende concretizar, centra-se unicamente nos aspectos de controlo de acesso a uma célula de inspecção, para controlo da qualidade de produtos semi-fabricados. A sua interligação com o ambiente envolvente será simulada no PC, pela geração a pedido do utilizador, de todos os acontecimentos relevantes.

2 Definição do Problema

A planta referente à área fabril onde se encontra implantada a célula de inspecção para controlo de qualidade pode ser observada na Figura 2. Um tapete rolante transporta para a célula os produtos a inspeccionar, ou retira para uma célula de recondicionamento, situada a montante no processo produtivo, os produtos que não conseguiram passar no controlo de qualidade aqui realizado. A operação deste tapete é perfeitamente transparente, face ao problema que pretendemos resolver.

Na verdade, as operações que pretendemos supervionar dizem apenas respeito aos produtos que tendo sido sujeitos a aprovação, irão ser colocados em *palletes* na plataforma de carga, por acção explícita do próprio inspector. A capacidade da plataforma de carga é de **quatro palletes**.

As *palletes* são transportados desta plataforma para a células de fabrico situadas a jusante no processo produtivo, por intermédio de filoguiados (AGVs[1]). Cada filoguiado tem capacidade para transportar uma **única pallete**. Em cada momento, **apenas um** filoguiado, pode aceder à plataforma de carga.

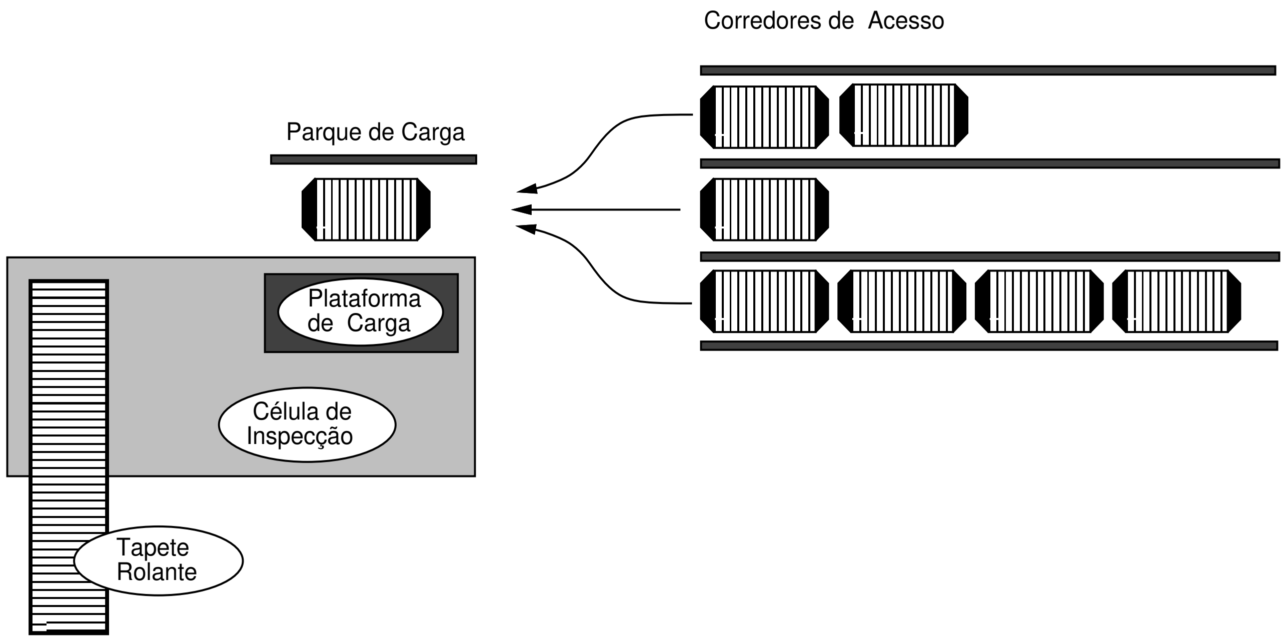

Figura 2: Planta da célula fabril a controlar

Precedendo a plataforma de carga, existem **três corredores** de acesso, onde os filoguiados aguardam a sua oportunidade de poderem recolher a correspondente *pallete*. Considere que a capacidade máxima de cada corredor é de **4 filoguiados**, mas assuma igualmente que a entidade responsável pelo seu encaminhamento **nunca** envia um filoguiado para um corredor cuja lotação esteja completa[2].

[1] *Automatic Guided Vehicles.*

[2] Isto significa que o seu programa **não deve** incluir mecanismos para controlo duma eventual violação das capacidades dos corredores, simplesmente porque ela nunca poderá ocorrer.

Cada um dos filoguiados que chega à célula de inspecção "transporta" uma *ordem de produção* que, para além de conter a identificação do filoguiado, dá a conhecer ao sistema as seguintes informações:

- **Tipo**, que identifica o corredor onde o filoguiado deve esperar caso não consiga de imediato acesso à plataforma de carga. Valores possíveis: a, b, c.
- **Prioridade** - nível de urgência associado a cada requisição de uma *pallete*. Valores possíveis: 100 a 125, correspondendo o primeiro destes valores ao nível mais urgente.

O acesso à plataforma de carga por parte dos filoguiados presentes nos corredores de acesso é governado pelas seguintes regras:

- Apenas **um** filoguiado pode estar estacionado junto à plataforma de carga, para recolha de uma *pallete*.
- Um filoguiado que não consiga de imediato acesso à plataforma de carga, por esta já se encontrar ocupada, deve aguardar a satisfação do seu pedido em fila de espera no respectivo corredor.
- As filas de espera associadas a cada corredor obedecem a uma filosofia **FIFO**, reflectindo o facto de não ser fisicamente possível realizar uma ultrapassagem, dentro de cada corredor.
- Por forma a prevenir um crescimento desequilibrado de alguma destas filas de espera, a ordem para um filoguiado se encaminhar para a plataforma de carga deverá ser preferencialmente concedida ao filoguiado que se encontra à cabeça da fila de espera com maior número de elementos.
- Em caso de empate, o serviço deve ser entregue ao filoguiado presente no corredor com o identificador mais baixo[3].
- As duas regras anteriores devem, em princípio, deixar de ser obedecidas sempre que nalgum dos corredores exista um **filoguiado crítico**. Um filoguiado é classificado como crítico quando "transporta" uma ordem de produção com nível de urgência superior a um dado limiar (que passaremos a designar por **prioridade crítica**).
- A classificação do filoguiado (crítico/normal) **pode ser alterada** caso o valor da prioridade crítica seja entretanto modificado.
- Existindo filoguiados classificados como críticos, o serviço terá de ser obrigatoriamente entregue a um corredor onde um destes filoguiados se encontre estacionado. A posição dos filoguiados críticos no interior de cada corredor de acesso irá condicionar a eleição da fila de espera de onde sairá o próximo filoguiado a movimentar-se. Em concreto, impõe-se que:
  - o serviço deva ser entregue preferencialmente ao corredor onde o número de filoguiados que precedem o primeiro dos filoguiados críticos[4] é menor.
  - em caso de empate deverá, de entre os corredores empatados, escolher-se aquele onde a distância[5] para o próximo filoguiado crítico é menor. Esta metodologia de escolha deverá ser aplicada recursivamente até à resolução do empate.

[3] Assumindo que a ordenação crescente é a, b, c.
[4] Isto é, o mais próximo da saída.
[5] Medida em número de filoguiados, a partir do filoguiado crítico actual.

- caso o empate subsista até não haverem mais filoguiados críticos nestes corredores, a escolha será efectuada aplicando as regras usadas quando não existem filoguiados críticos (i.e. escolha pela ocupação dos corredores ou, em caso de empate, pela ordem dos seus identificadores) aos corredores que permanecem empatados.

- A tomada de decisão sobre a entrega do serviço é sempre realizada no momento em que um filoguiado acabou de abandonar a plataforma de carga. O filoguiado escolhido deverá então avançar imediatamente para o parque da plataforma de carga.

- Na concretização do seu algoritmo considere que os tempos de movimentação e de recolha da *pallete* por parte dos filoguiados não é desprezável (mais adiante lhe diremos como deve modelar estas variáveis).

Pretende-se concretizar um esquema de controlo que coordene o acesso dos diferentes filoguiados ao parque da plataforma de carga. Para além disso, deve ser possível visualizar (no écran do PC) o estado da célula a controlar, no que toca aos seguintes parâmetros:

- Acompanhamento do filoguiado escalonado enquanto este se dirige para o parque da plataforma de carga.
- Sinalização do filoguiado que se encontra a aguardar/carregar a *pallete*.
- Identificadores dos inspectores que aprovaram as *palletes* a carregar/carregada.
- Acompanhamento do filoguiado que abandona a plataforma de carga, após recolha de uma *pallete*.
- Representação das *ordens de produção* associadas aos filoguiados estacionados nos corredores de acesso, bem como a sinalização de filoguiados críticos.

3 Desenvolvimento do Trabalho

O trabalho a realizar poderá dividir-se nos seguintes aspectos:

- Concepção de um algoritmo para controlo de acesso dos filoguiados ao parque da plataforma de carga, obedecendo aos requisitos anteriormente enunciados, e correspondente programação utilizando as facilidades do E286.
- Desenvolvimento de procedimentos que permitam visualizar (no écran do PC) a evolução do sistema. Este tipo de funcionalidade é usualmente conhecida por **sinóptico**.
- Concepção e programação, como parte integrante da aplicação, dum servidor de pedidos do utilizador.
- Concepção e programação dos procedimentos de interacção entre uma tarefa monitora[6] (já fornecida) e o servidor de pedidos.

Estes componentes deverão ser construídos de uma forma modular, não necessariamente pela ordem indicada, com interacções mínimas entre si. Esclarecemos seguidamente os aspectos mais relevantes do trabalho a desenvolver.

[6] Que irá ter como missão gerar determinados acontecimentos (introdução de um filoguiado no sistema, colocação de *palletes* na plataforma de carga, modificação da prioridade crítica.).

3.1 Os Filoguiados

Cada filoguiado deverá ser representado a nível de sistema como uma tarefa do E286 (de prioridade igual à atribuída ao filoguiado). Estas tarefas deverão ser criadas exclusivamente pelo servidor de pedidos, sob solicitação explícita do utilizador.

As operações a efectuar por cada uma daquelas tarefas são essencialmente as mesmas, sendo apenas distinguíveis em termos da sua representação externa (**sinóptico**) pelos parâmetros próprios de cada filoguiado (identificador, prioridade e tipo). Os últimos parâmetros são indicados explicitamente pelo utilizador, enquanto o primeiro é gerado pelo servidor de pedidos (veja secção 3.4).

Pelas razões acima indicadas, todos os processos devem partilhar o **mesmo segmento de código**, devendo os seus parâmetros individuais ser-lhe passados da forma que os alunos entenderem mais conveniente. Em [2] discutem-se diferentes métodos utilizáveis na passagem de parâmetros para o interior duma tarefa.

Os tempos associados à movimentação dos filoguiados e à duração da operação de recolha da *pallete* deverão ser modelados através das funções `agvMove` e `agvPick` (veja apêndice B). Após permanecer um período dado por `agvTurn` fora da célula de inspecção, admitimos que um filoguiado volta aqui "transportando" uma *ordem de produção* idêntica.

3.2 O Algoritmo de Controlo

A programação do algoritmo de controlo deve ser construída a partir das primitivas disponíveis na **interface utilizador do E286**. A solução a adoptar deverá ser eminentemente distribuída, com as diferentes tarefas a cooperarem entre si. De forma alguma se admitem soluções de controlo centralizado.

Espera-se vivamente que os alunos utilizem racionalmente os serviços proporcionados pelo E286, explorando as suas potencialidades no sentido de diminuírem a complexidade dos algoritmos a concretizar e simplificarem a sua programação. Uma compreensão adequada das noções relativas à criticalidade da multi-tarefa e ao respectivo esquema de escalonamento por prioridades, numa arquitectura actuando com concorrência e paralelismo em tempo-real, são aqui particularmente relevantes.

De igual forma, uma correcta estruturação e utilização da informação que descreve o estado e a arquitectura da aplicação é imprescindível para a simplificação do trabalho de programação.

3.3 Procedimentos de Visualização

A parte esquerda do écran do PC deverá ser integralmente dedicada à apresentação do **sinóptico** da aplicação, tal como ilustramos na Figura 3. A função principal do sinóptico é a de reportar a localização dos filoguiados (corredores, parque da plataforma de carga, movimentação na célula de inspecção ou operação noutras células de fabrico) pela visualização das respectivas *ordens de produção*, contendo: identificador do filoguiado, prioridade e tipo. Um filoguiado crítico deverá ainda sinalizar esta sua condição pela impressão dum asterisco (*) adicional. O valor actual do nível de urgência (ou prioridade) crítico deverá ser igualmente reportado na janela reservada para esse efeito.

A parte inferior do sinóptico destina-se à representação dos filoguiados que por não terem conseguido acesso imediato à plataforma de carga, terão de aguardar pela sua vez no corredor, associado ao respectivo tipo. A representação dos filoguiados nestas zonas do écran deve reproduzir fielmente a sua ordem de chegada. Assim no topo de cada uma das janela deverão encontrar-se os filoguiados que,

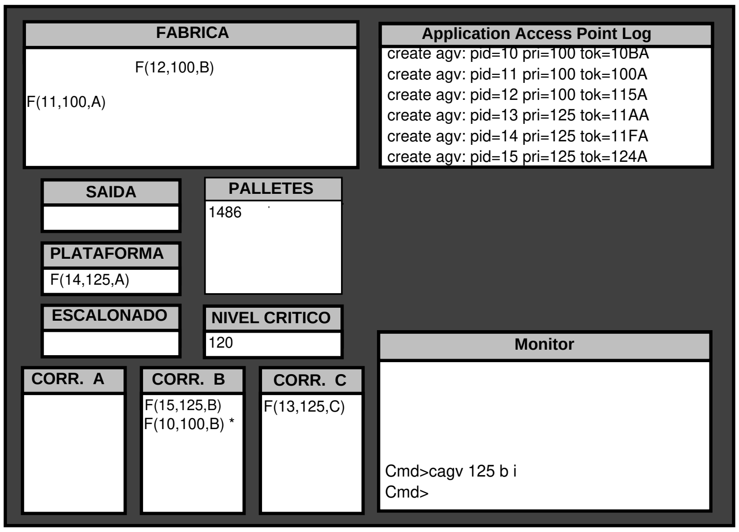

Figura 3: Exemplo dum écran de visualização.

dentro de cada fila, há mais tempo aguardam uma oportunidade de acesso. Quando o filoguiado que se encontra nestas circunstâncias finalmente recebe autorização para passar à plataforma de carga, todos os outros que se mantêm nesse corredor devem ser promovidos uma posição. Por exemplo: quando o filoguiado 15 conseguir acesso à plataforma de carga, o filoguiado 10 deverá ocupar a primeira posição da janela CORR B (Figura 3).

Na situação ilustrada na Figura 3, a plataforma de carga é ocupada pelo filoguiado 14, que se encontra a recolher uma *pallete* aprovada pelo inspector 1486. Ao terminar esta operação, o filoguiado deve movimentar-se para a saída, devendo a sua ordem de produção passar a ser representada na janela correspondente, até a sua saída efectivamente se consumar. Durante este período a identificação do inspector que aprovou a *pallete* transportada deve manter-se afixada. (Será eliminada quando o filoguiado sair definitivamente da célula de inspecção.)

Após sair da célula de inspecção, o filoguiado deverá passar a ser representado na janela FABRICA, na coluna correspondente ao seu tipo e na primeira linha que encontrar livre, sinalizando a sua localização noutras células de fabrico. Nesta zona a classificação do filoguiado como crítico/normal não assume qualquer significado, pelo que a representação da ordem de produção nunca deve incluir tal informação.

A saída do filoguiado irá ainda eleger o próximo filoguiado a dirigir-se para a plataforma de carga. No caso da Figura 3, será eleito o filoguiado 15, devendo a sua ordem de produção passar a ser visualizada na zona ESCALONADO, enquanto durar a sua movimentação para a plataforma de carga. A chegada do filoguiado à plataforma, faz com que a sua ordem de produção deixe de ser visualizada na janela ESCALONADO, para o passar a ser na janela PLATAFORMA.

O sinóptico deverá ser construído de forma modular, exportando as funções `synPrintState` e `synCleanState`, para serem utilizadas pelos filoguiados. A interface destas funções, que **não deve ser alterada**, encontra-se especificada no ficheiro correspondente, apresentado no apêndice B.

Completando a descrição do écran de visualização, iremos encontrar no seu lado direito a janela privada do servidor de pedidos, onde a actividade desta tarefa é reportada, e a janela correspondente ao monitor. A zona compreendida entre estas duas janelas poderá ser ocupada por uma janela de diagnóstico, para onde serão ser reportados eventuais mensagens de diagnóstico e de controlo de erros. Uma janela de diagnóstico definida por `dbgScreenOpen` permanece invisível, até à impressão da primeira mensagem.

3.4 O Servidor de Pedidos

A função exclusiva do servidor de pedidos é a de fornecer um ponto de entrada único e perfeitamente identificado para a aplicação de aquisição de dados[7].

Esta tarefa deverá obrigatoriamente estar bloqueada num objecto do E286 para comunicação entre tarefas (fila de caracteres, por exemplo). Ao receber um pedido, o servidor deve interpretá-lo e proceder à sua execução. Os pedidos a servir envolvem as seguintes operações: introdução dum novo filoguiado no sistema; aprovação duma *pallete*; alteração da prioridade crítica.

O servidor de pedidos deverá ser responsável pela geração do *identificador único* do filoguiado. Exige-se que o primeiro identificador gerado tenha o valor 10.

3.5 O Monitor

A tarefa monitora tem por função suportar o diálogo com o utilizador para a realização dum conjunto diversificado de operações, das quais as mais importantes se destinam a simular a geração de acontecimentos relacionados com a evolução da célula.

Esta aplicação, que já se encontra quase integralmente concretizada, consiste num interpretador interactivo de comandos, que proporciona algumas facilidades rudimentares de edição. Após ler e interpretar a linha de comando fornecida a partir do teclado, o monitor ou executa internamente o comando ou chama uma função externa (a definir pelos alunos no ficheiro `comando.c` - apêndice B).

Todas as funções do monitor são evocadas de forma congruente com a semântica habitual das interfaces aos sistemas operativos, em que os parâmetros de evocação especificam o número de argumentos (`argc`) e um vector de argumentos (`argv`). Nos comandos a concretizar pelos alunos, os argumentos das respectivas funções deverão ser interpretados e validados, procedendo-se posteriormente à formação duma mensagem a enviar ao servidor de pedidos. Os comandos disponíveis e/ou a concretizar são os seguintes:

sos - Proporciona uma ajuda rudimentar.

cls - Limpa o écran do monitor.

dosboot - Promove o re-arranque do MS-DOS. Não se garante o correcto funcionamento deste comando em todos os PCs.

cagv <pri> <type> <sta> - Permite criar um processo, especificando a sua prioridade (entre 100 e 125, inclusivé) e tipo (a, b, c). O último parâmetro indica a localização do filoguiado no momento da sua criação. Valores possíveis: **i** (célula de inspecção) ou **f** (outras células da fábrica).

aprov <id> - Comando fornecido por um inspector (de identificador **id**) para aprovação duma *pallete* e sua disponibilização na plataforma de carga. O campo **id** é constituído por quatro dígitos.

chcri <level> - Permite alterar o nível da prioridade crítica.

3.6 Diversos

- As excepções "hardware" não são mascaradas pelo núcleo tendo como consequência a "destruição" da respectiva tarefa. O caso mais frequente resulta dos mecanismos de protecção de

[7]Usualmente designado por *ponto de acesso ao serviço*.

memória: o uso de ponteiros inválidos gerará imediatamente uma *excepção*.

- Algoritmos de sincronização mal concebidos podem levar a situações de impasse (interblocagem). A utilização de mensagens de diagnóstico pode ajudar a detectar estes casos.
- Uma vez que cada tarefa deverá possuir uma zona de dados própria, a partilha de dados requer o uso de segmentos partilhados: use as funções do núcleo para o efeito.

4 Entrega do Trabalho

Este trabalho deve ser entregue em data e local $\boxed{\text{A INDICAR}}$. O material a entregar consiste numa *diskette* com o programa desenvolvido (fontes, objectos e makefile) e numa listagem do mesmo. Todos estes elementos devem vir identificados com o número do grupo de laboratório e identificação dos alunos que o compõem, devidamente acondicionados numa capa apropriada. A visualização posterior do trabalho será efectuada com base no trabalho entregue.

5 Perguntas e respostas

P 1 *Como posso instalar o E286 no meu PC ?*

Consulte o apêndice A.

P 2 *Tentei instalar o E286 no meu PC e não consegui. Existe algum procedimento que não tenha sido divulgado?*

Não. O E286 foi instalado no laboratório usando exactamente os procedimentos que foram indicados. Verifique todos os passos que efectuou. Veja P 3.

P 3 *O código carrega nos PC's com vírus?*

Pelo menos, não carrega nos PC's infectados com o "stoned". Seja cuidadoso. Use só disquetes "desinfectadas". Utilize uma *diskette* com o sistema e com uma versão recente de anti-vírus: carregue o sistema a partir da disquete e remova os vírus.

P 4 *Devo simplificar sempre que possível?*

Sim. Simplifique os algoritmos e a programação o mais que puder, respeitando o requisitos do enunciado. Opte sempre pelo que der menos trabalho.

P 5 *Tenho liberdade para acrescentar ao trabalho funcionalidades adicionais ?*

Nem pensar! Uma eventual alteração à especificação contida no enunciado deve ser previamente discutida com o respectivo assistente.

O corpo docente reserva o direito de penalizar (até um máximo de **3 valores**) a introdução de aspectos funcionais totalmente ortogonais à natureza do trabalho. Exemplos: écrans sofisticados para indicação da composição do grupo, anúncios à *Coca-Cola*, etc.

P 6 *O E286 tem "bugs"?*

As "queues", "semaphores" e "regions" **foram testadas** e **estão a funcionar correctamente**. As "mailboxes" e os portos de comutação de ordens, estão temporariamente indisponíveis.

P 7 *Existe algum limite para o número de tarefas?*

Sim, mas este é suficiente para as exigências do trabalho. Corra o programa de demonstração **tlimits** para conhecer o valor deste limite.

P 8 *Existe diferença entre ponteiros "far" e "near" nas chamadas ao E286?*

Sim. Quando é pedido um ponteiro, este deve ser um ponteiro do tipo "far". A única excepção a esta regra consiste no ponteiro para o erro, o qual deve ser "near". Exemplo:

```
static word   erro;
static char   c;
       word   nbytes;
       token  queue;
...
nbytes = rq_receive_queue ( queue, (char far *)&c, 1, FOREVER, &erro );
...
```

P 9 *Para aceder/copiar dados de outro segmento tenho que usar necessariamente o "movedata"?*

Não, pode gerar um ponteiro "far", usando o token devolvido ao criar o segmento. Por exemplo:

```
token     seg;
byte far  *p;
...
seg = rq_create_segment   ( SZ_SEG, &err );
p   = (byte far *) makedw ( seg, 0 );
```

P 10 *Tentei criar um ciclo infinito com a instrução "while (1) {}" e deu erro. É normal?*

Nem todos os compiladores geram um ciclo infinito neste caso. Use outra coisa qualquer, p.e. "while (1) $i++$;"

P 11 *O tamanho do segmento usado na tarefa inicial é suficiente para armazenar todas as contantes e variáveis globais?*

A configuração actual define para o segmento de código uma dimensão aproximadamente igual a 32K e um segmento de dados de 16K, o que deverá ser suficiente! Caso contrário, a aplicação poderá ser reconfigurada (um processo bastante complexo) para aumentar a dimensão do segmento em causa. Lembre-se que as cadeias de caracteres usadas dentro dos "printf" são memorizadas no segmento de dados da tarefa principal. Se usar essas chamadas noutras tarefas, a zona de dados copiada deverá abranger estas constantes. As extensões ao núcleo `rq_task_execv` e `rq_task_execl`, já realizam esta operação automaticamente.

Referências

[1] José Alves Marques and Paulo Guedes. *Fundamentos de Sistemas Operativos*. Editorial Presença, 1990.

[2] José Rufino. *Introdução à Programação de Aplicações Concorrentes no E286*. IST/INESC, 1995.

[3] Carlos Sebastião and Telmo Carmo. *E286: Manual do Utilizador*. INESC.

A Instalação do E286

Este apêndice descreve os procedimentos a observar na instalação do executivo multi-tarefa de tempo-real E286 num PC.

1. Antes de iniciar o processo de instalação deve munir-se duma cópia actualizada da *diskette* de distribuição do E286.

   > **Contacte pessoalmente o seu assistente para obter este pacote de *software*. Não o copie directamente a partir dos discos dos PCs do laboratório.**

2. Colocando-se em A: dê o comando "`instala`". Como resultado, irá ser criada uma árvore de directórios debaixo de "`c:\e286`". Consulte o ficheiro "`c:\e286\leia.me`" para obter pormenores adicionais referentes ao conteúdo e função de cada um dos sub-directórios criados pelo processo de instalação.

3. Antes de prosseguir com a instalação do E286 assegure-se que já instalou o compilador MSC.

   > **É obrigatória a utilização do compilador da Microsoft MSC 6.00, no desenvolvimento deste trabalho.**

4. Altere o "`config.sys`" de forma a desactivar eventuais gestores de memória que tenha instalados, nomeadamente o EMM386.

5. Inclua no seu "`autoexec.bat`" os comandos especificados no ficheiro "`c:\e286\autoexe.cmd`".

   > **Em relação aos dois pontos anteriores, adopte uma configuração o mais simples possível para o seu PC, enquanto trabalhar com o E286.**

6. Tem agora de configurar o E286 de acordo com o tipo de écran. A maioria das placas gráficas actuais funciona correctamente com uma configuração policromática (a fornecida por omissão), independentemente do tipo de écran. Se tal não for o caso, o comando "`c:\e286\bin\ecbw`" permite estabelecer uma configuração monocromática. O comando "`c:\e286\bin\eccolor`" repõe a configuração policromática.

7. Carregue de novo o `DOS` (faça "*reset*") de modo a tornar efectivas as alterações anteriores.

8. Coloque-se no directório "`c:\e286\work`". Deverá copiar para aqui o conteúdo do directório "`..\examples\`" ou "`..\sources\`", consoante pretenda compilar algum dos ficheiros de exemplo ou o embrião da aplicação a desenvolver. De o comando:

   (a) "`nmake nome_do_exemplo`", no primeiro caso ou simplesmente
   (b) "`nmake`", no segundo caso.

   > **O processo de geração de aplicações através de *makefiles* assenta na consistência do grupo data/hora dos ficheiros. Por isso verifique/acerte a data do seu PC antes de iniciar qualquer sessão de trabalho.**

   A *makefile* irá gerar os ficheiros necessários ao carregamento da aplicação em memória. No final do procedimento de ligação de objectos o "`linker`" emite o aviso "`no stack segment`". Não se trata dum erro.

9. Execute agora o comando "`eloadpc`". O E286 deve carregar e lançar a aplicação anteriormente gerada. Para voltar ao DOS é necessário fazer *reset* ao computador ou dar o comando "`dosboot`", no caso da aplicação fornecida em "`..\sources\`".

> **Atenção: A concretização do comando "`dosboot`" é extremamente dependente da arquitectura do PC. Por isso não se assegura o seu correcto funcionamento em todos os PCs.**

B Embrião da Aplicação a Desenvolver

No directório `c:\e286\sources\` são fornecidos um conjunto de ficheiros fonte, que constituem o embrião da aplicação a desenvolver. Os ficheiros que concretizam o monitor (`monitor.c`), as funções gráficas (`tfvideo.[ch]`) e a sequência de arranque (`aplmain.c`) poderão apenas necessitar de modificações de pormenor. A parte fundamental do trabalho deverá ser editada nos restantes ficheiros procurando, sempre que possível, respeitar a estruturação aí definida.

Os ficheiros a que nos estamos a referir são:

- `aplserv.[ch]` - Onde se definem os procedimentos de arranque da aplicação e o servidor de pedidos.
- `filogui.[ch]` - Onde irão ser definidas as acções a realizar pelos filoguiados.
- `synoptc.[ch]` - Onde se definem os procedimentos associados ao sinóptico da aplicação.
- `comando.c` - Onde se definem as funções chamadas pelo monitor, para a execução dos comandos pedidos neste enunciado e ainda não concretizados.

Seguidamente apresentam-se as listagens de todos os ficheiros anteriormente mencionados, tal como são fornecidos no disco de distribuição do E286.