



Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Licenciatura em Engenharia Informática e de Computadores

SISTEMAS OPERATIVOS (SI-10/11)

3º Trabalho Prático – Gestão de Memória

Objectivo:

Desenvolvimento de aplicações no sistema Operativo Windows NT/2000/XP/Vista, usando a Win32 API; Construção de bibliotecas de ligação dinâmica (DLL); Tratamento de excepções estruturadas (SEH) da WIN32. Análise e desenvolvimento de aplicações WIN32 com recurso às primitivas da API de gestão de memória e gestão de memória virtual.

Análise dos exemplos

Analise os exemplos disponíveis na página da disciplina sobre utilização de excepções SEH, DLL e Gestão de memória da WIN32 (figura 1).

4 Exemplos

- 01-ListArgs
- 02-TratamentoErros
- 03-CriacaoProcessos
- 04-CriacaoTarefas
- 05.1-Sincronismo
- 05.2-Exemplos Clássicos sobre Sincronismo
- 05.3-Exemplos de Sincronismo na API Win32
- 06.1-Demos GUI
- 06.2-Demos GUI - Create Child Window Control
- 06.3-Demos GUI - CAD
- 07.1-DLL-TLS-Random
- 07.2-DLL-TLS-extrasV2
- 08-Excepcoes-SEH
- 09.1-GestaoMemoriaVirtual
- 09.2-GestaoMemoriaVirtual - FileMapping

Figura 1 – Exemplos referentes às DLL's, tls, excepções SEH e Gestão de Memória na WIN32

1ª Parte – Exercício prático

Considere, para efeitos da questão, que está a desenvolver um módulo a ser integrado numa aplicação e que disponibiliza um vector¹ (*array*) de elevadas dimensões (por exemplo até às dezenas de Megs) para efeitos do registo de histórico de informação (log). No entanto, este vector só é necessário ocasionalmente e as suas exigências, em termos de dimensão, variam podendo ir das centenas de bytes até ao valor máximo. **Assim, não faz sentido que a aplicação ocupe (*allocate*), inicialmente, a totalidade da dimensão**, mas sim ir atribuindo recursos físicos à medida das necessidades. O vector para o registo de informação é da exclusiva propriedade de uma tarefa podendo esta, caso necessite, iniciar um vector deste tipo. Utilizando as primitivas de gestão de memória virtual da WIN32, o tratamento de excepções estruturadas (SEH) e o mecanismo *Thread Local Storage* (TLS) para associar dados a uma tarefa específica, desenvolva uma biblioteca dinâmica (DLL) que ofereça a funcionalidade acima descrita.

As funcionalidades a disponibilizar pelo módulo são:

- Uma função para a iniciação do suporte do registo de histórico de informação, da tarefa que invoca a função, onde deve ser recebido a dimensão máxima admitida (`CreateThreadLog(...)`);
- Uma função para adicionar informação ao registo de histórico (`AppendThreadLog(...)`);
- Uma função para que permita libertar um bloco de informação mais antigo (segundo a lógica de um buffer circular), e.g. `FreeThreadLog(..., nBytes)` o que significa que o espaço dos `nBytes` mais antigos é libertado;
- Uma função para libertar todos os recursos ocupados pelo registo de histórico (`ResetThreadLog(...)`);
- Uma função para libertar o suporte do registo de histórico (`DestroyThreadLog(...)`).

Propõe-se como metodologia de realização deste trabalho a sua divisão pelas seguintes etapas:

1. Implemente as funções `CreateThreadLog`, `AppendThreadLog`, `FreeThreadLog`, `ResetThreadLog`, `DestroyThreadLog` considerando que apenas existe um vector de histórico (vector global) de forma a testar a implementação das funções no que diz respeito à reserva de memória e utilização das SEH para atribuição de memória à medida das necessidades.

¹ Esta noção de vector assume que a disposição em memória é contígua e a indexação pode ser realizada de forma semelhante aos tradicionais Arrays das linguagens de programação.

2. Realize um pequeno programa de teste que permita avaliar o correcto funcionamento das funções anteriores. Faça testes que levem ao armazenamento de volumes avultados de informação. Utilize o programa VMMap, apresentado nas aulas, para verificar a atribuição de memória.
3. **[Opcional – 4 valores]** Altere as funções desenvolvidas no ponto anterior para que o vector seja associado à tarefa que utiliza as funções. Assim cada tarefa pode possuir um vector de histórico. Neste ponto vai introduzir na implementação a utilização do mecanismo de TLS.
4. Realize um pequeno programa de teste que permita avaliar o correcto funcionamento do ponto anterior.
5. **[Opcional – 3 valores]** Organize as funções de suporte ao registo de histórico numa biblioteca dinâmica (DLL). Utilize os programas utilitários dumpbin ou depends, apresentados nas aulas, para verificar os símbolos exportados pela DLL e os símbolos importados pelo módulo executável.
6. **[Opcional – 3 valores]** Considere a possibilidade das funções CreateThreadLog, AppendThreadLog, FreeThreadLog, ResetThreadLog, DestroyThreadLog poderem reportar os erros através das excepções SEH.

As alíneas opcionais da metodologia sugerida, para a realização deste trabalho, possuem esta indicação juntamente com o valor que contribuem para a nota deste trabalho.

Nota:

Na resolução desta questão não pode tirar partido das funções de gestão de *heaps* disponíveis na Win32 API, nomeadamente, `HeapCreate(...)`, `HeapAlloc(...)` e `HeapFree(...)`.

2ª Parte – Exercícios Teóricos

1. Na figura 2 apresenta-se o código de uma função (UmaFuncao()) implementada com recurso às exceções SEH. Analise o seu código e determine o resultado devolvido, após a sua execução, descrevendo e justificando os passos da sua execução.
2. Considere uma arquitectura com suporte à gestão de memória virtual através de paginação com uma estrutura de quatro níveis. Sabendo que a dimensão das páginas é de 4KB, cada tabela de suporte à paginação dos 4 níveis ocupa uma página e que a dimensão das entradas das tabelas é de 8 Bytes indique:
 - O esboço da organização deste sistema de gestão de memória;
 - A dimensão do espaço de endereçamento virtual;
 - O número de páginas existentes no espaço de endereçamento virtual;
 - A dimensão, mínima e máxima, ocupada pelas estruturas de gestão de memória virtuais associadas a cada processo.
3. Indique, justificando convenientemente as suas afirmações, as implicações na adopção de páginas de maior dimensão, por exemplo 8KB, na gestão de memória virtual baseada em paginação.
4. Na gestão de memória virtual com paginação que tipos de fragmentação ocorrem no espaço de endereçamento virtual e no espaço de endereçamento físico? Considere as situações em que um programa requisita várias zonas de memória (páginas) com a função VirtualAlloc e que em algumas situações não utiliza, completamente, uma página.
5. Compare a utilização das funções VirtualAlloc e HeapAlloc disponibilizados na Win32 API.

```
DWORD UmaFuncao() {  
    DWORD dwTemp = 0;  
  
    while (dwTemp < 10) {  
  
        __try {  
            if (dwTemp == 2)  
                continue;  
  
            if (dwTemp == 3)  
                break;  
        }  
        __finally {  
            dwTemp++;  
        }  
  
        dwTemp++;  
    }  
  
    dwTemp += 10;  
    return(dwTemp);  
}
```

Figura 2

Entrega do trabalho de grupo

A entrega deverá ser feita até ao dia **7 de Janeiro de 2011**. Considerando a existência da concentração deste trabalho com os últimos trabalhos de outras disciplinas o trabalho poderá ser entregue, impreterivelmente, até a data de 2ª época (**Não serão aceites trabalhos após esta data**).

A entrega do trabalho **realiza-se, exclusivamente, na página da turma no Moodle**. A entrega do trabalho é constituída pelo relatório, onde lista e explica a sua solução e resultados observados, e as soluções do *Visual Studio* para que possam ser testadas na discussão.

Nas directorias das soluções do *Visual Studio* 2008 existe um ficheiro com a extensão *ncb* (que contém informação de suporte ao *Intellisense*) que deve ser eliminado de forma a reduzir a dimensão da solução a ser submetida. No caso de utilizar o *Visual Studio* 2010 deve eliminar o ficheiro com extensão *sdf* e a directoria *ipch*. Deve, igualmente, eliminar o conteúdo das directorias *Debug* e/ou *Release* da sua solução.

Todos os elementos que compõem o trabalho (relatório, código, etc.) deverão ser entregues num ficheiro comprimido do tipo *Zip* ou *Rar*. O relatório deverá ser entregue no formato *pdf*².

Bom trabalho,

Nuno Oliveira

² Poderá criar ficheiros em pdf com o seguinte utilitário grátis: <http://www.primopdf.com/>