



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

1º Trabalho Prático – Processos/Tarefas

Objectivo:

Desenvolvimento de aplicações no sistema Operativo Windows NT/2000/XP/Vista, usando a API Win32; análise e desenvolvimento de aplicações recorrendo a multiprocessamento tirando conclusões, relativamente, a vantagens/desvantagens em diversos cenários de execução. Escrita e código independente do tipo de codificação de caracteres utilizado (ASCII/Unicode).

Análise dos exemplos

Analise os exemplos disponíveis na página da disciplina, no moodle (ver figura 1), sobre a criação de processos e de tarefas.



Figura 1 – Exemplos referentes à criação de processos e tarefas

Utilize a ferramenta *Process Explorer*¹ apresentada na figura 2 (disponível na página comum da disciplina) para analisar os recursos utilizados por cada aplicação, como por exemplo, número de threads, prioridade dos processo e tarefas, handles, e módulos constituintes de um processo, etc.

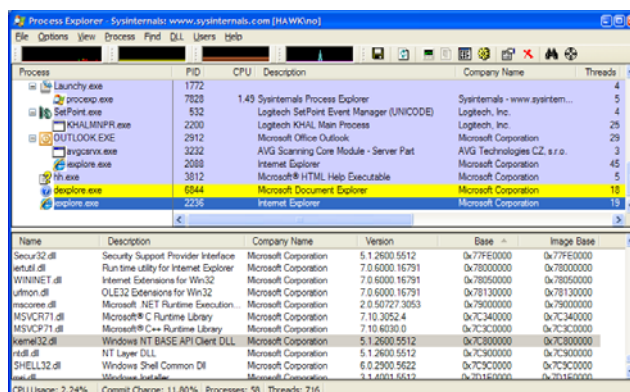


Figura 2 - Ferramenta *Process Explorer* da Sysinternals

¹ www.sysinternals.com

Como complemento ao estudo e preparação para a realização do trabalho prático sugerem-se as seguintes alterações aos exemplos:

- Altere o exemplo 01-LancaProcessosConsola para que o mesmo passe a mostrar na consola os tempos associados à execução dos processos filhos, nomeadamente, o tempo de início, fim, dispendido em modo protegido e modo utilizador.
- Altere o exemplo 02-MyHelloWorldThreads de modo a permitir receber nos seus argumentos o número de tarefas a lançar e o tempo de execução que estas deverão demorar (assuma que o tempo de execução é efectuado através da chamada de sistema `Sleep(DWORD)`).
- Experimente executar o exemplo 02-MyHelloWorldThreads através do exemplo 01-LancaProcessosConsola.

Análise à execução de processos e tarefas e o seu escalonamento

Pretende-se que crie três aplicações que representem o espectro de aplicações que se executam num sistema operativo. Uma aplicação de computação intensiva, outra de I/O intensivo e outra híbrida em que alterna períodos de computação e de I/O.

Pretende-se que efectue diversos testes que considere representativos de forma a poder comprovar e concluir como é realizado o escalonamento, dos fios de execução, no Windows.

Alguns dos factores que poderá analisar podem ser, por exemplo, o nível de prioridade e o ajustamento dinâmico que é efectuado pelo Windows, os tempos de CPU gastos pelas tarefas em *kernel mode* e *user mode*.

Uma ferramenta que poderá utilizar para registar o histórico dos valores que seleccionar poderá ser o “Performance Monitor”, exemplificado na figura 3, que poderá encontrar em Control Panel → Administrative Tools → Reliability and Performance Monitor (ou performance no caso Windows XP).

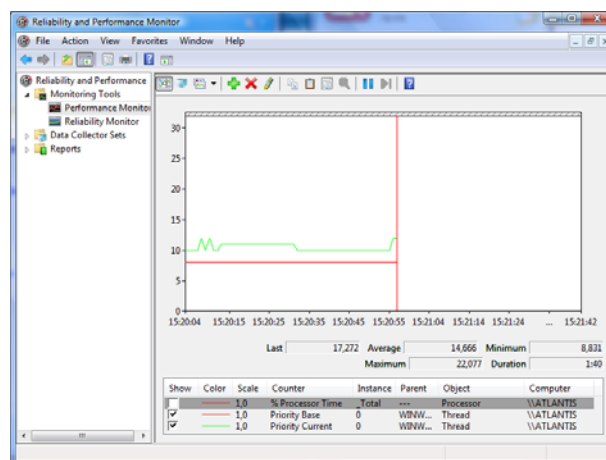


Figura 3 – Performance Monitor permite a análise da evolução de prioridade de uma tarefa.

Tendo por base este objectivo, realize uma aplicação que auxilie o utilizador a simular estes cenários de execução e a respectiva monitorização das suas características com base nos três tipos programas anteriores

(computação intensivo, I/O intensivo ou híbrido). Assim, esta aplicação deve apresentar ao utilizador um menu permitindo-lhe a escolha do tipo de programa a executar. A aplicação deve então, permitir a execução dos vários tipos de programas e de várias instâncias do mesmo programa. Para além disso devem ser possíveis as seguintes opções:

- a) Listar o estado actual de execução de cada processo criado pela aplicação (em execução, terminado)
- b) Alterar a prioridade dos processos lançados pela aplicação
- c) Listar a informação (thread ID e prioridade) sobre todas as tarefas existentes num processo lançado pela aplicação. Nota: pesquise na documentação “*traversing thread list*”.
- d) Alterar a prioridade da tarefa principal de um processo lançado pela aplicação. Genericamente, pode considerar a hipótese de alterar a prioridade de qualquer tarefa com o apoio da alínea anterior.
- e) Suspender a execução de um processo (realizar `SuspendThread` a todas as tarefas do processo)
- f) Terminar as tarefas de um processo
- g) Terminar o processo

Cálculo do número PI com múltiplas tarefas

Pretende-se uma aplicação Win32, em modo consola, com múltiplas tarefas para determinar o valor do número PI. O método a utilizar baseia-se na geração de pontos aleatórios (número de experiências) contidos num quadrado de dimensão unitária contabilizando o número de pontos que pertencem ao maior círculo inscrito no quadrado. Sabendo que a área do círculo (πr^2) é igual a $\pi/4$ para um quadrado com $L=1$ em que a área ($L \times L$) é igual a 1 tem-se que a razão entre o número de pontos que pertencem ao círculo e o número total de experiências é um valor aproximado a $\pi/4$.

Uma estratégia a seguir consiste na existência de várias tarefas dividindo o número de experiências entre si. Assim, cada tarefa gera pontos aleatórios e contabiliza o número de pontos contidos no círculo. Depois de todas as tarefas terminarem somam-se os contadores parciais de cada tarefa e divide-se pelo número de experiências obtendo assim um valor aproximado a $\pi/4$.

Considere a possibilidade de indicar à aplicação, através da linha de comando, o número de tarefas e o número de experiências a utilizar. Na figura 4 seguinte mostra-se um exemplo de cálculo com base em duas tarefas e um total de 400 mil experiências.

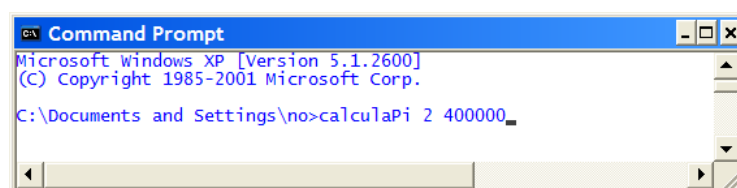


Figura 4 – Exemplo da invocação do programa de cálculo do valor de PI com 400 mil experiências

Avalie o tempo total de execução variando o número de experiências e o número de tarefas. Experimente obter resultados, executando a aplicação, em diferentes arquiteturas *hardware*. Na apresentação dos

resultados, não se esqueça de identificar as várias arquiteturas utilizadas nos ensaios, como por exemplo, o tipo de processador.

Se alterar as prioridades das tarefas do cálculo do PI que implicações existem tanto ao nível do cálculo como ao nível de todo o sistema

NOTA: para obter os tempos de execução poderá recorrer às primitivas: `QueryPerformanceCounter`, `GetTickCount`, `timeGetTime`.

2ª Parte – Exercícios Teóricos

1. Indique, justificando, em que consiste um Sistema Operativo e quais as suas responsabilidades.
2. Uma aplicação não tem acesso directo para manipular o *hardware*, por exemplo, uma placa de som. Especifique como é possível a uma aplicação utilizar estes recursos.
3. Indique, justificando, em que consistem as instruções privilegiadas e de que modo contribuem para garantir o correcto funcionamento do Sistema Operativo.
4. O sistema operativo multitarefa possui múltiplas tarefas em competição pelo(s) CPU('s). A decisão de qual a próxima tarefa a executar-se é responsabilidade do escalonador (*scheduler*). Num sistema que utilize um escalonamento com base no nível de prioridade das tarefas indique o que entende por *preemptive scheduling* e em que consiste o conceito de *starvation*.
5. Considere um cenário num sistema operativo onde foram submetidos o seguinte conjunto de processos para execução. A tabela seguinte indica o tempo em milissegundos do tempo necessário de CPU (CPU burst) e da prioridade atribuída a cada processo (Assuma que os valores menores são os mais prioritários).

Processo	<i>Burst Time</i>	Prioridade
<i>P1</i>	10	3
<i>P2</i>	1	1
<i>P3</i>	2	3
<i>P4</i>	1	4
<i>P5</i>	5	2

Admitindo a ordem de submissão dos processos *P1*, *P2*, *P3*, *P4*, *P5* responda às seguintes questões:

- a) Desenhe os diagramas temporais da execução destes processos se forem adoptados os algoritmos de escalonamento FCFS, SJF, algoritmo prioritário *non-preemptive* e RR com *time quantum* de 1 unidade de tempo.
- b) Qual o tempo total de execução (*turnaround*) por processo para cada um dos algoritmos de escalonamento utilizados na alínea a).
- c) Qual o tempo médio de *Turnaround* em cada um dos algoritmos?
- d) Qual o tempo de espera (*Waiting time*) por processo para cada um dos algoritmos de escalonamento utilizados na alínea a).
- e) Qual o tempo médio de espera em cada um dos algoritmos.

Entrega do trabalho de grupo

A entrega deverá ser feita até ao dia **10 de Maio de 2009**. Por cada dia de atraso na entrega do trabalho/relatório, este terá uma penalização de **0,4 Valores**.

A entrega do trabalho contempla o relatório, onde lista e explica a sua solução e resultados observados, e as *solutions* do *visual studio* para que possam ser testadas na discussão.

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.

A entrega dos trabalhos será efectuada na página da turma no Moodle.

Está disponível na página comum da disciplina³ um fórum para a colocação de dúvidas frequentes (FAQ).

Bom trabalho,

Diogo Remédios e Nuno Oliveira

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

³ <http://moodle.isel.ipl.pt/deetc/course/view.php?id=853>