

# Teste de Sistemas Operativos

## Correcção do Grupo I

fsm

Junho 2016

### Resumo

Pretende-se com este documento ajudar a compreender a forma correcta de responder ao Grupo I do teste de Sistemas Operativos (SO), permitindo uma melhor preparação para os alunos que terão de comparecer ao exame de recurso. A intenção não é dar aqui soluções completas, "replicáveis" às cegas em qualquer outra prova de avaliação. A ideia é ficar a perceber os caminhos errados, quer de interpretação de enunciado quer de estratégia de resposta. Quando chegar ao fim do texto deverá ser capaz de, por si só, chegar à resposta correcta.

## 1 Antes de começar...

As provas de avaliação destinam-se a mostrar os resultados da aprendizagem. Aprendizagem. Compreender novos problemas, equacionar soluções, optar e saber justificar essa escolha, implementar ou descrever a solução, conforme o caso, e finalmente demonstrar que funciona no caso concreto que foi apresentado. Em SO reproduzir um "padrão" vale muito pouco. Não é o resultado que interessa, mesmo que possa estar correcto. Primeiro há que provar que é trabalho / aprendizagem pessoal e ser-se capaz de descrever o que faz e que benefício traz este algoritmo, esta estratégia, o uso desta arquitectura ou system call, etc.

A palavra de ordem é justificar. Respostas descritivas, sem justificação e compromisso pessoal são equivalentes à antiga "morte súbita" do futebol: acabou o jogo por aqui. Derrota. <sup>1</sup>.

## 2 Primeira Questão

### 2.1 Enunciado

*No sistema de backup/restore do trabalho prático deste ano havia clientes, vários servidores, sinais e 1 FIFO. Que justificação tem esta aparente complexidade para algo que na sua essência é cópiar de ficheiros? Escolha dois objectivos desta abordagem, os que lhe parecerem mais importantes, e demonstre que sem esta arquitectura esses objectivos dificilmente seriam atingidos. Pretende-se uma resposta sucinta, menos de 10 linhas legíveis.*

### 2.2 Discussão

E se o enunciado tivesse sido o seguinte?

*Se observar os bares da UMinho verifica que existem clientes, um balcão e vários funcionários. Que justificação tem esta aparente complexidade para algo que na sua essência é pegar e consumir um artigo aí fornecido? Escolha dois objectivos desta abordagem, os que lhe parecerem mais importantes, e demonstre que sem esta arquitectura esses objectivos dificilmente seriam atingidos. Pretende-se uma resposta sucinta, menos de 10 linhas legíveis.*

Mantém a resposta que deu no teste? Que resposta daria agora?  
E se a sua resposta tivesse sido uma destas?

---

<sup>1</sup>Se assistiu ao jogo Portugal-Islândia, no final teve um exemplo claríssimo deste comportamento. Quando se quis saber a opinião do treinador sobre as causas do empate para se aprender com ele e não o repetir, este limitou-se a relatar o jogo... Não soube justificar. Falou mas não respondeu à pergunta. Nota negativa. E não falemos então da Áustria

- *Sim, na verdade nos bares da UMinho existem clientes, um balcão e vários funcionários. O objectivo é comunicação. Esta aparente complexidade para algo que na sua essência é pegar e consumir um artigo fornecido justifica-se porque sem isso seria muito difícil termos clientes a comunicar com os vários funcionários.*

Morte súbita! Não percebeu que comunicação é uma consequência, não um objectivo. Tendo optado pelo modelo cliente/servidor, é óbvio que o cliente tem de comunicar ao servidor o que pretende!!! E o facto de repetir textualmente as palavras do enunciado, sabendo que só tem apenas 10 linhas para responder, é muito mau presságio.

- *Sim, na verdade nos bares da UMinho existem clientes, um balcão e vários funcionários. Esta aparente complexidade para algo que na sua essência é pegar e consumir um artigo fornecido justifica-se porque sem balcão não havia funcionários.*

Morte súbita! À pergunta "porquê a existência de separação entre clientes e funcionários" responde com "sem balcão não havia funcionários"... Algo que estaria correcto se tivesse realmente percebido que há bares que funcionam sem funcionários, são as pessoas/clientes que retiram os artigos. A pergunta número 2 do grupo I diz exactamente isso!!! Até pede para descrever e justificar o algoritmo dos clientes quando não há balcão nem funcionários.

- *Sim, na verdade nos bares da UMinho existem clientes, um balcão e vários funcionários. Esta aparente complexidade para algo que na sua essência é pegar e consumir um artigo fornecido justifica-se porque um dos objectivos é a concorrência e sem funcionários não havia concorrência.*

Morte súbita! Concorrência há sempre (em sistemas actuais, Linux, Windows, Android, IOS...). Esses sistemas operativos gerem alegremente centenas de processos concorrentes, e não é preciso ter servidores para isso. No r/c do edifício do DI há bares (máquinas) sem funcionários e nos intervalos das aulas há mesmo muita concorrência de utilizadores. E no guião 2 havia pipes e servidores? Claro que não? E no guião 3? No 4? Servidores só apareceram no Guião 6 é que apareceu a arquitectura cliente/servidor, antes fazia-se o mesmo, por vezes com algumas limitações, mas fazia-se. Onde está esse conhecimento?

- *Sim, na verdade nos bares da UMinho existem clientes, um balcão e vários funcionários. Esta aparente complexidade para algo que na sua essência é pegar e consumir um artigo fornecido justifica-se porque um dos objectivos é a segurança ou privacidade e sem funcionários e balcão a separar não havia segurança.*

Rasgo de esperança mas vai morrer na praia! Continua-se a responder com o enunciado mas já se refere algo que pode estar no bom caminho. No entanto, são frases soltas, sem nenhuma justificação, o que é inaceitável para quem fez um trabalho. Ou devia ter feito, em vez de se limitar a saber descrever o que alguém fez. É preciso provar que realmente há segurança (o que quer que isso seja) e que sem a arquitectura cliente/servidor, isto é, só com a cultura dos guiões até aos pipes/FIFOs não se teria segurança. Sem essa prova, nada feito. E

- *Sim, na verdade nos bares da UMinho existem clientes,... Valerá a pena mostrar as 160+ respostas seguintes? Há até quem depois de repetir o enunciado não consiga escrever mais nada.*

Lembra-se de uma aula inteira, 2 horas em que o professor fez a análise do trabalho prático começando por recordar o que é e para que serve um backup, dizendo que se a ideia fosse só salvar-guardar os ficheiros bastaria a cópia destes, de preferência para um dispositivo diferente (porquê?) mas que neste trabalho há também o requisito da eficiência de espaço que se consegue através da compactação e detecção de duplicados? Falou nessas 3 fases, cópia, compactação e cálculo da assinatura do ficheiro para detecção de duplicados. Insistiu que backup implica cópia, porque o objectivo não é poupar espaço (alguns ingénuos até moveram ficheiros no trabalho prático... deviam ter tido logo aí morte súbita a SO) mas sim ter uma cópia para poder recuperar esse estado caso haja algum problema sério com o ficheiro inicial. Backup não é archive.

Depois falou-se dos utilizadores usarem o comando `sobucli` para fazer pedidos a um processo servidor e esse sim, recebia os pedidos, iria criar mais processos para executar o `gzip` e `shasum`? E falou de benefícios gerais da arquitectura cliente/servidor face à alternativa do cliente ter de fazer

a operação completa e nesse caso tomar precauções se estiver a executar em paralelo com outros clientes?

Que benefícios tem então a arquitectura cliente/servidor? Cuidado, se for ao Google ou Wikipédia perguntar pelos benefícios, tenha muita atenção porque não é um caso qualquer. É o de um backup. E não é um backup qualquer, o tal que "essencialmente é uma cópia de ficheiros" (e que por isso normalmente não precisaria de arquitectura cliente/servidor<sup>2</sup>). É um backup eficiente, com compressão e deduplicação!

Mas não é então o servidor quem faz o serviço e o cliente só pede e espera pelo sinal de resposta? Isso não é para ter segurança? Nem por isso, pelo menos num sistema em que há só uma máquina e o sistema de ficheiros é partilhado. Lamento dar a má notícia que não é o balcão nem o funcionário que dão segurança. Quem tiver um braço muito comprido (ou uma cana de pesca, stick de selfie, etc.) consegue chegar aos artigos à guarda do servidor! E porquê?

...Pense! ...

A segurança, mais especificamente privacidade, negação de acesso a outros utilizadores, não se faz com um balcão ou FIFO mas sim colocando os backups num local inacessível. I-N-A-C-E-S-S-Í-V-E-L. Primeiro há que impedir os clientes de chegarem aos bolos, deles ou dos outros, não interessa. Não lhes tocam. Se quiserem pedem ao funcionário para lhos guardar e devolver<sup>3</sup>. Inacessível, é isso. E como se consegue? Resposta: da mesma forma que se consegue sem cliente/servidor... Da mesma forma que há coisas que não consegue/não sabe fazer e alguém lhe ensinou a fazer `sudo`.

Ah!!!! Com permissões! Então vou ter a pergunta certa?

Calma. É verdade que a acessibilidade a objectos no sistema de ficheiros se faz usando permissões. Mas... azar, para não morrer na praia não basta "mencionar" permissões. Há que dizer como é que um ficheiro meu (é o meu casaco, no fim continuo a poder usá-lo, certo?) é emprestado ao funcionário e lá já não o posso usar. É aqui que entram os utilizadores diferentes, pois as permissões em Linux são divididas em owner, group e others, o dono, o grupo e os outros. Deve ser claro que o servidor tem de ser um "user" diferente do dono do ficheiro original, e agora vem a frase chave, para que nem o dono nem ninguém além do funcionário seja capaz de alterar as permissões com `chmod`.

Então se não tenho 100% pelo menos tenho 80%?

Acho que ainda não... Se a ideia da privacidade é tornar inacessíveis os meus backups aos outros utilizadores, então ainda consigo fazer isso sem cliente/servidor, usando apenas tecnologia dos guiões 1, 2 e 3. Fizemos isso no guião 1 quando se criou o ficheiro de 10MB. Houve problemas com a `UMASK`, gente que não conseguia aceder ao ficheiro que tinha acabado de criar com as permissões erradas. Se a isso juntar uma pasta de backup invisível (`.backup`), que eu desconheço mas é usada por um comando que me limito a usar, então só digo `sobackup *.c` e tenho backup...

Em resumo, o enunciado mencionava 2 benefícios, o primeiro seria privacidade mas teria de ser respondido falando nas permissões e utilizadores diferentes para impedir `chmod`, e aí sim, "pode entrar" a arquitectura C/S. Pode? Então ainda não tenho essa parte certa? Não, então ainda no parágrafo anterior se mostrou que também se consegue privacidade sem cliente/servidor!!! Então essa pergunta é traiçoeira e ninguém pode tê-la completamente certa? Também não, há um caso em que só com cliente/servidor se consegue o que se pretende. Leia de novo o enunciado do trabalho prático.

O segundo benefício não é concorrência porque (a) concorrência implica custos que se evitam se eu próprio puder tirar o artigo da máquina e (b) em SO há concorrência mesmo sem arquitectura cliente/servidor (Guiões 1-5). Há de facto uma segunda razão para a arquitectura ser cliente/servidor, provavelmente só ao alcance de quem pensou muito no trabalho e a cada passo que deu, a cada decisão que tomou, se interrogou como faria isso sem cliente/servidor.

As perguntas 2 e 3 seguem dentro de momentos...

---

<sup>2</sup>Porque aparece aqui a palavra "normalmente"? Então não é sempre?

<sup>3</sup>Nem aqui somos originais, há tantos sítios onde há serviço de guarda de objectos pessoais, por exemplo um casaco ou capacete, em restaurantes, no Marés Vivas, etc.