

Trabalho prático de Sistemas Operativos

Processamento de Notebooks

Grupo de Sistemas Distribuídos
Universidade do Minho

Abril de 2018

Nota prévia

Este trabalho prático vale 1/3 da classificação final e serve essencialmente para averiguar o grau de preparação (individual) para o teste e/ou exame de recurso, onde não vai existir o auxílio externo nem dos restantes elementos do grupo. Espera-se que seja bastante simples e rápido de realizar para quem foi resolvendo os exercícios propostos nos guiões. Quem não os fez tem ainda muito tempo para recuperar, aproveitando as aulas teóricas e práticas até à data da entrega ou marcando sessões adicionais de esclarecimento de dúvidas.

1 Resumo

Neste trabalho pretende-se construir um sistema para processamento de *notebooks*, que misturam fragmentos de código, resultados da execução, e documentação. Neste contexto, um notebook é um ficheiro de texto que depois de processado é modificado de modo a incorporar resultados da execução de código ou comandos nele embebidos. Como exemplo, um ficheiro `exemplo.nb` com o conteúdo:

```
Este comando lista os ficheiros:
$ ls
Agora podemos ordenar estes ficheiros:
$| sort
E escolher o primeiro:
$| head -1
```

Este ficheiro ao ser processado, leva a que as linhas começadas por `$` sejam interpretadas como comandos que serão executados, sendo o resultado produzido inserido imediatamente a seguir, delimitado por `>>>` e `<<<`. As linhas começadas por `$|` executam comandos que têm como *stdin* o resultado do comando anterior. O ficheiro ficará então com o conteúdo sendo algo como:

```
Este comando lista os ficheiros:
$ ls
>>>
coisa.c
a.out
batata.c
<<<
Agora podemos ordenar estes ficheiros:
```

```

$| sort
>>>
a.out
batata.c
coisa.c
<<<
E escolher o primeiro:
$| head -1
>>>
a.out
<<<

```

2 Processador de notebooks

O processador de notebooks deverá ser um comando que dado um nome de ficheiro, interpreta-o, executa os comandos nele embebidos, e acrescenta o resultado da execução dos comandos ao ficheiro.

2.1 Funcionalidades básicas

2.1.1 Execução de programas

Como descrito, as linhas começadas por `$` devem ser interpretadas como comandos (programa e argumentos), e se começadas por `$|` o comando deve ter como *stdin* o resultado do comando anterior. Em qualquer caso, o resultado produzido deve ser inserido no ficheiro, imediatamente a seguir ao comando respectivo, delimitado por `>>>` e `<<<`. O conteúdo assim produzido e delimitado, deverá ser ignorado (não interpretado como comandos).

2.1.2 Re-processamento de um notebook

Depois de processado pela primeira vez, um notebook poderá ser editado pelo utilizador, que poderá modificar os comandos a executar, e mais tarde processado novamente. O processador deverá então substituir os resultados do processamento anterior pelos novos resultados.

2.1.3 Detecção de erros e interrupção da execução

Caso algum dos comandos não consiga ser executado, não termine com sucesso, ou escreva algo para o *stderr*, o processamento deve ser abortado, devendo o notebook deve ficar inalterado.

O utilizador também deverá poder interromper um processamento em curso, com `^C`, devendo o notebook permanecer inalterado.

2.2 Funcionalidades avançadas

2.2.1 Acesso a resultados de comandos anteriores arbitrários

Os comandos deverão poder ser generalizados para terem como *stdin* o resultado o *n*-ésimo comando anterior, com linhas começadas por `$n|`, por exemplo `$4|` para se referir ao quarto comando anterior, sendo `$1|` equivalente a `$|`.

Esta funcionalidade deverá ser obtida evitando executar vários vezes o mesmo comando.

2.2.2 Execução de conjuntos de comandos

Depois de ter a funcionar a execução de 1 comando e respectivos argumentos (veja os exemplos da secção Resumo), poderá acrescentar pipelines ou até pensar em execução concorrente. Se decidirem avançar para a execução em background, por favor falem antes com os docentes. Não é nada fácil e poderá ser um mau compromisso em termos de (acréscimo de classificação)/(tempo dispendido).

Informações adicionais

- Conforme indicado no início do ano, o método de avaliação de Sistemas Operativos contempla a realização e discussão de um trabalho prático onde é necessário ter nota mínima de 10 valores. Quem teve nota positiva no ano imediatamente anterior pode optar por manter a nota anterior (truncada a 15). Basta que esses alunos verifiquem no Blackboard que fazem parte da lista dos que não precisam de entregar novo trabalho prático. Se não estiverem na lista (disponível no início da próxima semana), agradeço que enviem mail para fsm@di.uminho.pt . Se estiverem, não necessitam de fazer mais nada.
- Cada grupo de trabalho deve ser constituído por até três elementos. Grupos de dimensão inferior a três terão de ser justificados e autorizados pelos docentes. Todos os elementos do grupo terão de estar à vontade para responder às questões colocadas durante a discussão pública, justificando as opções tomadas e mostrando conhecimento do código desenvolvido, dos testes efetuados, da configuração do sistema, etc. Pretende-se que cada um mostre o que aprendeu e que estará à vontade no teste/exame.
- Para quem tem de fazer trabalho prático, serão disponibilizados no blackboard muitos grupos (LCC 25, MIEI 100) com capacidade para 3 elementos. Quando o grupo estiver formado, APENAS UM dos elementos escolhe um grupo que esteja vazio e inscreve-se nesse grupo. De seguida comunica aos restantes elementos o número do grupo em que se inscreveu para que os seus colegas se registem também nesse grupo. Essa inscrição irá estar disponível durante cerca de duas semanas, encerrando de seguida as inscrições a modo a permitir o planeamento dos horários das apresentações.
- O trabalho deve ser entregue até às 23:59 do dia 2 de Junho, através do Blackboard. Deve ser enviado um único ficheiro em formato ZIP, contendo um pequeno relatório em PDF, o código fonte, a makefile para compilação de todo o código desenvolvido e eventualmente um script para configuração inicial. Será essa a versão que irão demonstrar no dia da discussão¹.
- A avaliação dos trabalhos deverá ocorrer nos dias 4 a 6 de Junho. Imediatamente após contagem do número de trabalhos submetidos será publicado no blackboard um conjunto de horários (em princípio 1/2 hora por grupo). Cada um destes grupos do Blackboard corresponderá a um determinado slot de 1/2 hora, sendo necessária a presença de TODOS os elementos desse grupo de alunos.

¹Toda a gente tem de saber instalar e correr o software sem comprometer o horário de avaliação. Para evitar percalços e atrasos no setup da demonstração do trabalho, a demonstração poderá ser pedida a qualquer dos elementos do grupo enquanto outros vão explicando o código e as decisões tomadas. Poderá ser realizada num computador qualquer, seja de um dos profs ou colega de grupo. Por este motivo, não deve colocar caminhos absolutos no sistema de ficheiros, por exemplo `open ("/Users/fsm/...")`. Use caminhos relativos. E não se esqueça de colocar na makefile a possibilidade de remoção de tudo o que foi criado na demonstração. Obrigada.