

Laboratórios de Sistemas Operativos

Tutorial #3: Programação com ficheiros

Os tutoriais práticos de SO consistem num conjunto de exercícios práticos que permitem aos alunos familiarizarem-se com um determinado tema que será necessário para resolver os projetos da disciplina. Os tutoriais podem ser resolvidos individualmente ou em grupo. A sua resolução é recomendada mas não obrigatória. Não são avaliados.

Cada tutorial pressupõe que os exercícios são realizados numa interface de linha de comandos (shell) de um sistema Unix/Linux ou equivalente. Assume também que os alunos já resolveram os tutoriais anteriores.

Acesso a ficheiros

1. Descomprima *open-write.zip* e observe o código fonte. Compile, execute e verifique que o ficheiro *test.txt* tem a mensagem esperada.
2. Descomprima agora *open-read.zip* e observe o código fonte. Compile, execute e verifique que o output impresso no ecrã é o esperado.
3. Modifique cada programa acima para passarem a aceder ao ficheiro usando as funções da biblioteca *stdio* (ou seja, *fopen*, *fwrite*, *fread*, *fclose*, entre outras).
4. Experimente correr os novos programas e confirme que têm um comportamento equivalente aos exemplos originais.
5. Que vantagens encontra nas novas variantes dos programas?
6. Caso o programa *open-read.c* seja executado sobre um ficheiro cujo conteúdo exceda 128 bytes, só um excerto do ficheiro é lido. Corrija essa limitação, de forma a ler integralmente o ficheiro mesmo quando o seu conteúdo é superior à dimensão do *buffer*.
7. Teste a sua solução usando um ficheiro grande.
8. Estenda a solução anterior de modo que o resultado da leitura seja colocado no ficheiro *'test-out.txt'*.
9. Experimente a nova solução e verifique o resultado está correto usando o comando *diff* entre ambos os ficheiros.

Interface de Entradas/Saídas

A interface de E/S Unix é baseada na existência de uma tabela de ficheiros abertos por cada processo, também chamada *file descriptor table*. Esta tabela permite tratar vários canais de comunicação da mesma

forma (ex. terminais, ficheiros, sockets, pipes) utilizando os *file descriptors* devolvidos pela função *open*.

Como já sabe, por convenção, os três primeiros file descriptors são reservados: 0 - *stdin* (*standard input*, vulgo teclado), 1 - *stdout* (*standard out*, vulgo *display*), 2 - *stderr* (*standard error*, vulgo *display*).

1. Modifique o programa original *open-write.c* substituindo o primeiro argumento (fd) da instrução *write* por '1' e verifique o que muda no comportamento do programa.
2. Também pode fazer esta alteração sobre o programa que compôs baseado na *stdio*. Nesse caso, deve passar a variável *FILE *stdout* (definida em *stdio*) para o argumento respetivo da função de escrita que usou (*fwrite*, *fprintf*, ou outra).
3. Modifique o programa original *open-read.c* substituindo o primeiro argumento (fd) da instrução *read* por '0' e verifique que a mensagem é lida do teclado.
4. Idem sobre o programa *open-read.c* baseado em *stdio*, usando a variável *FILE *stdin* como argumento da função de leitura (*fread*, *fscanf*, ou outra).