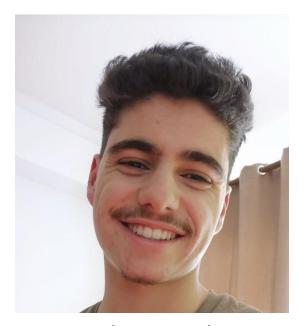
# Relatório

Rodrigo Ferreira Pintassilgo (2191190) e por Daniel Pires Patrício (2191195) declaram sob compromisso de honra que o presente trabalho (código, relatórios e afins) foi integralmente realizado por nós, sendo que as contribuições externas se encontram claramente e inequivocamente identificadas no próprio código. Mais se declara que os estudantes acima identificados não disponibilizaram o código ou partes dele a terceiros.



Rodrigo Pintassilgo



Daniel Patrício

# Signals:

- ✓ SIGINT: Totalmente operacional.
  - 1. Interrupção do teclado (CTRL + C).
  - 2. Com o printf() mostrou-se o sinal e o PID do processo que enviou o sinal à nanoShell.
  - 3. Término da nanoShell, através da função exit(0).

# ✓ SIGUSR1:

1. Não implementado.

# ✓ SIGUSR2:

1. Não implementado.

# • Opcão -f, --file <fich>:

- ✓ Totalmente operacional.
- ✓ Para a sua realização foram utilizadas as funções file\_exists(), file\_parameter().
- ✓ Quando é solicitado o parâmetro -f, é efetuada uma verificação, pois este parâmetro não é compatível com nenhum outro, assim (com esta verificação) é salvaguardado este caso.
- ✓ Através fa função file\_exists, verifica-se se o ficheiro dado como argumento do parâmetro -f, existe. Se não existir, dá erro, se existir chama a função file\_parameter.
- Esta função lê linha a linha os comandos, através de getline() conseguimos aceder a cada linha do ficheiro, a função strtok() permite dividir a string por espaços nos comandos. Assim, será possível executar os comandos presentes no ficheiro.

# • Opcão -h, --help:

- ✓ Totalmente implementado.
- ✓ É um menu de ajuda sucinta, ou seja, deu-se printf() de todas as opções de comandos, de forma a mostrar ajuda para cada uma das opções de comando. Contém também, a identificação dos estudantes que realizaram o projeto.

#### • Opcão -m, --max <int>:

- ✓ Totalmente implementado.
- ✓ Para a sua realização foram utilizadas as funções max\_parameter() e subfunction\_for\_nano\_max(), stdout\_stderr\_redirection(), remove spaces from string, remove spaces(), compare chr().
- ✓ Quando é solicitado o parâmetro -m, é efetuada uma verificação, pois este parâmetro não é compatível com nenhum outro, assim (com esta verificação) é salvaguardado este caso. Certificou-se também, que o argumento passado pelo parâmetro -m é positivo, caso seja 0 ou negativo mostra a devida mensagem de erro. Assim, o main chama a função max\_parameter().
- ✓ Esta função realiza um ciclo for, que vai determinar qual é o número máximo de comandos que se pode executar na nanoShell\$, indicado pelo parâmetro -m. Dentro do ciclo, é chamada a função subfunction\_for\_nano\_max().
- ✓ A subfunction\_for\_nano\_max() permite executar a nanoShell e introduzir os comandos desejados pelo utilizador, até que seja atingido o limite máximo ou então, seja colocado o comando bye na nanoShell\$. Verifica a existência de caracteres que não são suportados pela nanoShell\$, tais como o ?, \*, |, " e '. E também a presença dos símbolos >, >>, 2> e 2>>, que é feita através da função stdout\_stderr\_redirection(). Esta função chama a remove\_spaces\_from\_string que remove os espaços de uma determinada string. A função freopen() permite executar cada um dos comandos que contenham os símbolos de redirecionamento:

- >: Redireciona o stdout para um ficheiro e sobrescreve o seu conteúdo. No caso de não existir ficheiro, este é criado.
- >>: Redireciona o stdout para um ficheiro e acrescenta o seu conteúdo. No caso de não existir ficheiro, este é criado.
- **2>**: Redireciona o stderr para um ficheiro e sobrescreve o seu conteúdo. No caso de não existir ficheiro, este é criado.
- **2>>:** Redireciona o stderr para um ficheiro e acrescenta o seu conteúdo. No caso de não existir ficheiro, este é criado.

O comando é executado através do execvp, que substitui a imagem do processo atual por uma nova imagem do processo.

# • Opcão -s, --signalfile:

- ✓ Totalmente implementado.
- ✓ Para a sua realização foram utilizadas as funções signalfile\_parameter() e subfunction\_for\_nano\_max(), remove\_spaces(), compare\_chr().
- ✓ Quando é solicitado o parâmetro -s, o main chama a função signalfile\_parameter().
- ✓ A função cria, no diretório corrente, o ficheiro signal.txt, que contém os comandos para o envio de sinais ao nanoShell. Foi criada uma matriz de forma a guardar cada um dos 3 sinais (SIGINT, SIGUSR1, SIGUSR2). Com um ciclo for escreveu-se os comandos no ficheiro, através da função fprintf(). O PID do processo nanoShell, foi obtido por getpid().
- ✓ Foi chamada de novo a função subfunction\_for\_nano\_max(), que permite executar a nanoShell e introduzir os comandos desejados pelo utilizador, até que seja colocado o comando bye na nanoShell\$.
- ✓ A função remove\_spaces() serve para remover os espaços para quando se faz um comando com espaços à direita e/ou à esquerda.
- ✓ A função count\_chr() conta as vezes que um determinado char aparece dentro de uma string.

# • Sem parâmetros:

- ✓ Totalmente implementado.
- ✓ Para a sua realização foram utilizadas as funções executing\_nanoShell\$() e subfunction\_for\_nano\_max().
- ✓ Quando não é colocado qualquer opção o main irá chamar a função executing\_nanoShell\$(), que por sua vez irá chamar a função subfunction\_for\_nano\_max() (explicada anteriormente, ao detalhe), que permite executar os comandos pretendidos, até ser escrito o comando bye, ou ser enviado um sinal para tal.