

#!/bin/bash

#####  
# Description for the intranet check (one line, support Markdown syntax)  
# `fork` should not be called when a command does not exist

#####  
# The variable 'compare\_with\_sh' IS OPTIONNAL  
#  
# Uncomment the following line if you don't want the output of the shell  
# to be compared against the output of /bin/sh  
#  
# It can be useful when you want to check a builtin command that sh doesn't  
# implement  
# compare\_with\_sh=0

#####  
# The variable 'shell\_input' HAS TO BE DEFINED  
#  
# The content of this variable will be piped to the student's shell and to sh  
# as follows: "echo \$shell\_input | ./hsh"  
#  
# It can be empty and multiline  
shell\_input="hbtn"

#####  
# The variable 'shell\_params' IS OPTIONNAL  
#  
# The content of this variable will be passed to as the paramaters array to the  
# shell as follows: "./hsh \$shell\_params"  
#  
# It can be empty  
# shell\_params=""

#####  
# The function 'check\_setup' will be called BEFORE the execution of the shell  
# It allows you to set custom VARIABLES, prepare files, etc  
# If you want to set variables for the shell to use, be sure to export them,  
# since the shell will be launched in a subprocess  
#  
# Return value: Discarded  
function check\_setup()  
{  
 LTRACE\_ALLOWED\_FUNCTIONS\_FILE\_TMP="\${LTRACE\_ALLOWED\_FUNCTIONS\_FILE}\_dup"  
 \$CP \$LTRACE\_ALLOWED\_FUNCTIONS\_FILE \$LTRACE\_ALLOWED\_FUNCTIONS\_FILE\_TMP  
  
 \$CAT \$LTRACE\_ALLOWED\_FUNCTIONS\_FILE\_TMP | \$GREP -vw "fork" > \$LTRACE\_ALLOWED\_FUNCTIONS\_FILE  
  
 \$RM -f \$LTRACE\_ALLOWED\_FUNCTIONS\_FILE\_TMP  
  
 return 0  
}

#####  
# The function 'sh\_setup' will be called AFTER the execution of the students  
# shell, and BEFORE the execution of the real shell (sh)  
# It allows you to set custom VARIABLES, prepare files, etc  
# If you want to set variables for the shell to use, be sure to export them,  
# since the shell will be launched in a subprocess  
#  
# Return value: Discarded  
function sh\_setup()  
{  
 return 0  
}

#####  
# The function `check\_callback` will be called AFTER the execution of the shell  
# It allows you to clear VARIABLES, cleanup files, ...  
#  
# It is also possible to perform additionnal checks.  
# Here is a list of available variables:  
# STATUS -> Path to the file containing the exit status of the shell  
# OUTPUTFILE -> Path to the file containing the stdout of the shell  
# ERROR\_OUTPUTFILE -> Path to the file containing the stderr of the shell  
# EXPECTED\_STATUS -> Path to the file containing the exit status of sh  
# EXPECTED\_OUTPUTFILE -> Path to the file containing the stdout of sh  
# EXPECTED\_ERROR\_OUTPUTFILE -> Path to the file continaing the stderr of sh  
#  
# Parameters:  
# \$1 -> Status of the comparison with sh  
# 0 -> The output is the same as sh  
# 1 -> The output differs from sh  
#  
# Return value:  
# 0 -> Check succeed  
# 1 -> Check fails  
function check\_callback()  
{  
 status=\$1  
  
 return \$status  
}