

## FYD500 - Homework V

The first number of assignments for FYD500, an introduction to linux.

### Exercise 1 - Simple BASH scripts

- 1) Write a Bash script that displays the following information, the path to your home directory, the path of the shell that you are using and the total amount of physical memory as well as swap space.

```
#!/usr/bin/env bash
clear

mem=$(free -hm | awk 'NR==2 { print $3}')
swp=$(free -hm | awk 'NR==2 { print $3}')

printf "\$HOME\t\t=%s\n$SHELL\t\t=%s\nPhys. mem\t=%s\nSwap\t\t=%s\n" \
    "$(pwd)" ${SHELL} ${mem} ${swp}
```

### Exercise 2 - Scripts with options

Now make a copy of the script in Exercise 1 and rework the script so that it works with five optional arguments:

```
scriptname [-h] [-s] [-m] [-p] [--help]

#!/usr/bin/env bash

#
_msg_help="The script will print home path [-o], current shell[-s], free memory[-m], swap sp
_msg_empty="Please supply an input argument, see --help for all alternatives."
_msg_error="Error, please see --help for available flags."

# Unset parameters
unset h s m p help

## Assign mem and swp to free memory and swap
_mem=$(free -hm | awk 'NR==2 { print $3}')
_swp=$(free -hm | awk 'NR==3 { print $3}')
_home=$(pwd)
_csh=${SHELL}

# Get opts does not support long options i.e. --help

for arg in "$@"; do
    shift
    case "$arg" in
        "--home")    set -- "$@" "-o" ;;
```

```

        "--shell")  set -- "$@" "-s" ;;
        "--mem")    set -- "$@" "-m" ;;
        "--swap")   set -- "$@" "-w" ;;
        "--help")   set -- "$@" "-h" ;;
        *)          set -- "$@" "$arg"
    esac
done

# Check
_flags=1
while getopts "osmwh" opt
do
    case "$opt" in
        "o") printf "%-6s\t%-8s\n" "HOME:" "${_home}";;
        "s") printf "%-6s\t%-8s\n" "SHELL:" "${_csh}";;
        "m") printf "%-6s\t%-8s\n" "MEM:" "${_mem}";;
        "w") printf "%-6s\t%-8s\n" "SWAP:" "${_swp}";;
        "h") clear; echo ${_msg_help};;
        "") clear; printf "%s" "${_msg_empty}";exit 1;;
        "?") clear; printf "%s" "${_msg_error}";exit 2;;
    esac
done
shift $(expr $_flags - 1)
exit 0

```

### Exercise 3 - Expansion

Create three variables, VAR1, VAR2 and VAR3; initialize them with the values “thirteen”, “13” and “Happy Birthday”. Export VAR2

### Exercise 4 - Conditional statements

Use an if/then/elif/else construct to print information about a certain month and year.

```

#!/usr/bin/bash

POS=()
while [[ $# -gt 0 ]]
do
    key="$1"
    case $key in
        -m|--month)
            MONTH="$2"
            shift #shifted past argument
            shift #shifted past value

```

```

        ;;
        -y|--year)
            YEAR="$2"
            shift #shifted past argument
            shift #shifted past value
            ;;
        *)
            POS+=("$1")
            shift
            ;;
    esac
done

_nofDays=$(cal $MONTH $YEAR | awk '{if(NR>2) print $0}' | wc | awk '{print $2}')

if [[ -n $# ]]; then #if $1 is non-zero
    if [[ -n $MONTH ]] && [[ -n $YEAR ]]; then
        printf "%-20s\t%d\n" "Specified year:" $YEAR
        printf "%-20s\t%d\n" "Specified month:" $MONTH
        printf "%-20s\t%d\n" "Num. of days:" $_nofDays
    elif [[ -n $MONTH ]]; then
        echo "Month"
    elif [[ -n $YEAR ]]; then
        echo "Year"
    fi
else
    echo "Please specify a (y)ear or a (m)onth"
fi

Ok! Now do the same thing but with case statements.

#!/usr/bin/bash

POS=()
while [[ $# -gt 0 ]]
do
    key="$1"
    case $key in
        -m|--month)
            MONTH="$2"
            SW_CASE+='M'
            shift #shifted past argument
            shift #shifted past value
            ;;
        -y|--year)
            YEAR="$2"
            SW_CASE+="Y"
    esac
done

```

```

        shift #shifted past argument
        shift #shifted past value
        ;;
    *)
        POS+=("$1")
        shift
        ;;
    esac
done

if [[ -z $YEAR ]];then
    YEAR=$(date +%Y)
elif [[ -z $MONTH ]];then
    MONTH=$(date +%m)
fi

_nofDays=$(cal $MONTH $YEAR | awk '{if(NR>2) print $0}' | wc | awk '{print $2}')

case $SW_CASE in
    M)
        printf "%-20s\t%d\n" "Specified year:" $YEAR
        printf "%-20s\t%d\n" "Specified month:" $MONTH
        printf "%-20s\t%d\n" "Num. of days:" $_nofDays
        ;;
    Y)
        printf "%-20s\t%d\n" "Specified year:" $YEAR
        printf "%-20s\t%d\n" "Specified month:" $MONTH
        printf "%-20s\t%d\n" "Num. of days:" $_nofDays
        ;;
    YM)
        printf "%-20s\t%d\n" "Specified year:" $YEAR
        printf "%-20s\t%d\n" "Specified month:" $MONTH
        printf "%-20s\t%d\n" "Num. of days:" $_nofDays
        ;;
    *)
        echo "Please specify a (y)ear or a (m)onth"
        ;;
esac

```