

# Assignment - 5

**Explain this following bash script:**

```
#!/bin/bash
space_free=$( df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%//' )
case $space_free in
[1-5]*)
echo Plenty of disk space available
;;
[6-7]*)
echo There could be a problem in the near future
;;
8*)
echo Maybe we should look at clearing out old files
;;
9*)
echo We could have a serious problem on our hands soon
;;
*)
echo Something is not quite right here
;;
esac
```

This Bash script checks the available disk space on a system and provides different messages based on the percentage of free space.

Let's break it down:

1. ``#!/bin/bash``:

This line indicates that the script should be interpreted using Bash.

2. ``space_free=$( df -h | awk '{ print $5 }' | sort -n | tail -n 1 | sed 's/%//' )``:

This line uses a pipeline of commands to get the percentage of used disk space and stores it in the variable ``space_free``. Here's the breakdown:

- ``df -h``: Displays disk space usage in human-readable format.
- ``awk '{ print $5 }'``: Extracts the fifth column, which represents the percentage of used space.
- ``sort -n``: Sorts the percentages in ascending order.
- ``tail -n 1``: Retrieves the last (highest) percentage.
- ``sed 's/%//'``: Removes the '%' character from the percentage.

3. ``case $space_free in ... esac``:

This is a case statement that evaluates the value of ``$space_free`` and executes the corresponding block of code based on the pattern matching:

- ``[1-5]*``: If the percentage is in the range 1-5, it echoes "Plenty of disk space available."
- ``[6-7]*``: If the percentage is in the range 6-7, it echoes "There could be a problem in the near future."
- ``8*``: If the percentage starts with 8, it echoes "Maybe we should look at clearing out old files."
- ``9*``: If the percentage starts with 9, it echoes "We could have a serious problem on our hands soon."
- ``*``: For any other value, it echoes "Something is not quite right here."