```
nano system_monitor.zsh
sanatkumargupta@Sanats-MacBook-Air ~ % chmod +x system_monitor.zsh
sanatkumargupta@Sanats-MacBook-Air ~ % ./system_monitor.zsh
System Monitoring Report
------------------------
CPU Usage: 32.91%
Memory Usage: 2247 MB / 6320 MB (6.17%)
Disk Usage: 23%
Network Activity: In: 4.42007e+06 KB, Out: 263458 KB
In: 4.42007e+06 KB, Out: 263458 KB
In: 4.42007e+06 KB, Out: 263458 KB
------------------------
sanatkumargupta@Sanats-MacBook-Air ~ %
```

```
#!/bin/zsh

# Function to get CPU usage
get_cpu_usage() {
    local cpu_usage=$(top -l 1 | grep "CPU usage" | awk '{print $3 + $5}')
    echo "CPU Usage: $cpu_usage%"
}

# Function to get Memory usage
get_memory_usage() {
    local mem_used=$(vm_stat | grep 'Pages active' | awk '{print $3}' | sed 's/\.//')
    local mem_free=$(vm_stat | grep 'Pages free' | awk '{print $3}' | sed 's/\.//')
    local mem_total=$(vm_stat | grep 'Pages wired down' | awk '{print $4}' | sed 's/\.//')
    mem_total=$((mem_total + mem_used + mem_free))
    local mem_usage=$(echo "scale=2; ($mem_used + $mem_total)*4/1024/1024" | bc)
    echo "Memory Usage: $(($mem_used*4/1024)) MB / $((($mem_used + $mem_total)*4/1024))
MB ($mem_usage%)"
}

# Function to get Disk usage
get_disk_usage() {
    local disk_usage=$(df -h / | awk 'NR==2 {print $5}')
    echo "Disk Usage: $disk_usage"
}

# Function to get Network activity
get_network_activity() {
    local net_activity=$(netstat -ib | awk '/en0/{print "In: " $7/1024 " KB, Out: " $10/1024 " KB"}')
    echo "Network Activity: $net_activity"
}

# Main monitoring function
monitor_system() {
    echo "System Monitoring Report"
    echo "-------------------------"
    get_cpu_usage
    get_memory_usage
    get_disk_usage
    get_network_activity
    echo "-------------------------"
}

# Run the monitoring function
monitor_system
```

—--------------- EXPLANATION —--------------------

System Monitoring Script

This script provides a simple way to monitor your system's CPU, memory, disk usage, and network activity using `zsh` on macOS. It collects and displays these metrics in a report format.

Script Explanation

Shebang

```
#!/bin/zsh
```

The shebang (`#!`) specifies the path to the `zsh` interpreter. This tells the system to use `zsh` to execute the script.

Functions

1. Get CPU Usage

```
get_cpu_usage() {
    local cpu_usage=$(top -l 1 | grep "CPU usage" | awk '{print $3 +
$5}')
    echo "CPU Usage: $cpu_usage%"
}
```

- **top -l 1**: Runs the `top` command in batch mode and outputs one iteration of the report.
- **grep "CPU usage"**: Filters the output to lines containing "CPU usage".
- **awk '{print $3 + $5}'**: Uses `awk` to sum the user and system CPU usage percentages (assuming `top` output format).
- **echo "CPU Usage: $cpu_usage%"**: Prints the calculated CPU usage.

2. Get Memory Usage

```
get_memory_usage() {
    local mem_used=$(vm_stat | grep 'Pages active' | awk '{print $3}'
| sed 's/\.//')
    local mem_free=$(vm_stat | grep 'Pages free' | awk '{print $3}' |
sed 's/\.//')
    local mem_total=$(vm_stat | grep 'Pages wired down' | awk '{print
$4}' | sed 's/\.//')
    mem_total=$((mem_total + mem_used + mem_free))
    local mem_usage=$(echo "scale=2; ($mem_used +
$mem_total)*4/1024/1024" | bc)
    echo "Memory Usage: $(($mem_used*4/1024)) MB / $((($mem_used +
$mem_total)*4/1024)) MB ($mem_usage%)"
}
```

- **vm_stat**: Provides virtual memory statistics.
- **grep 'Pages active'**: Filters output for active memory pages.
- **awk '{print $3}'**: Extracts the number of active pages.
- **sed 's/.//'**: Removes the trailing period from the number.
- **Calculations**: Adds up memory pages (active, free, wired) and converts to MB.
- **bc**: Basic calculator for floating-point arithmetic.
- **echo**: Prints the memory usage.

3. Get Disk Usage

```
get_disk_usage() {
    local disk_usage=$(df -h / | awk 'NR==2 {print $5}')
    echo "Disk Usage: $disk_usage"
}
```

- **df -h /**: Displays disk space usage for the root directory in human-readable format.
- **awk 'NR==2 {print $5}'**: Extracts the usage percentage from the second line.
- **echo "Disk Usage: $disk_usage"**: Prints the disk usage.

4. Get Network Activity

```
get_network_activity() {
    local net_activity=$(netstat -ib | awk '/en0/{print "In: " $7/1024 " KB, Out: " $10/1024 " KB"}')
    echo "Network Activity: $net_activity"
}
```

- **netstat -ib**: Displays network statistics including interface information.
- **awk '/en0/{print "In: " $7/1024 " KB, Out: " $10/1024 " KB"}'**: Filters for `en0` interface and calculates KB from bytes for input and output.
- **echo "Network Activity: $net_activity"**: Prints the network activity.

Main Monitoring Function

```
monitor_system() {
    echo "System Monitoring Report"
    echo "-------------------------"
    get_cpu_usage
    get_memory_usage
    get_disk_usage
    get_network_activity
    echo "-------------------------"
}
```

- **echo "System Monitoring Report"**: Prints the report header.
- **Function Calls**: Executes each function to collect and print system metrics.
- **echo "-------------------------"**: Prints a separator.

Execution

```
# Run the monitoring function
```

```
monitor_system
```

This calls the `monitor_system` function to execute the monitoring tasks and print the report.

How to Use

1. **Save the Script**: Save the above script as `system_monitor.zsh`.
2. **Make Executable**: Run `chmod +x system_monitor.zsh` to make the script executable.
3. **Run the Script**: Execute the script by running `./system_monitor.zsh`.

Example Output

```
System Monitoring Report
-------------------------
CPU Usage: 32.91%
Memory Usage: 2247 MB / 6320 MB (6.17%)
Disk Usage: 23%
Network Activity: In: 4.42007e+06 KB, Out: 263458 KB
-------------------------.
```
