perplexity

Generate a Python script that extracts a current condition of the local host like CPU load (%), Memory load (%), and running system services, and saves this information on file with date and time stamp.

Here's a Python script that extracts CPU load, memory usage, and running services, then saves the information with a timestamp:

```
import psutil
import datetime
import os
import subprocess
import platform

def get_system_metrics():
    try:
        # Get timestamp for filename
        timestamp = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        filename = f"system_status_{timestamp}.txt"

        # Get CPU usage percentage
        cpu_load = psutil.cpu_percent(interval=1)

        # Get memory usage
        memory = psutil.virtual_memory()
        memory_load = memory.percent

        # Get running services (platform-specific)
        running_services = []
        system_os = platform.system()

        if system_os == "Windows":
            for service in psutil.win_service_iter():
                if service.status() == 'running':
                    running_services.append(service.name())
        elif system_os == "Linux":
            try:
                services = subprocess.check_output(
                    ['systemctl', 'list-units', '--type=service', '--state=running'],
                    text=True
                )
                running_services = [line.split()[0] for line in services.split('\n') if '
            except subprocess.CalledProcessError:
                running_services = ["Unable to retrieve services - requires root privileg

        # Create report content
```

```
        report = f"""System Status Report - {timestamp}
========================================

CPU Load: {cpu_load}%
Memory Usage: {memory_load}%

Running Services:
----------------
"""
        report += '\n'.join(running_services) if running_services else "No services infor

        # Save to file
        with open(filename, 'w') as f:
            f.write(report)

        print(f"System status report saved to: {os.path.abspath(filename)}")

    except ImportError:
        print("Error: psutil module not installed. Install with 'pip install psutil'")
    except Exception as e:
        print(f"An error occurred: {str(e)}")

if __name__ == "__main__":
    get_system_metrics()
```

Requirements:

1. Install the required package:

```
pip install psutil
```

Features:

- Creates timestamped report files (e.g., `system_status_2023-12-31_23-59-59.txt`)
- Works on both Windows and Linux
- Captures:
  - Current CPU load percentage
  - Memory usage percentage
  - List of running services
- Handles permission issues on Linux
- Provides full path to generated file

Sample Output File Content:

```
System Status Report - 2023-12-31_23-59-59
========================================

CPU Load: 12.5%
Memory Usage: 45.2%
```

```
Running Services:
----------------
accounts-daemon.service
apache2.service
cron.service
dbus.service
...
```

Notes:

- On Linux, service enumeration requires root privileges for complete information
- The script uses cross-platform compatible libraries (psutil)
- Service enumeration method differs between Windows and Linux
- Memory percentage calculation is handled automatically by psutil
- CPU measurement uses a 1-second interval for accurate reading

To run:

```
python system_monitor.py
```