



## Modify this script:"import re

```
def extract_critical_errors(log_file):
    with open(log_file, 'r') as file:
        for line in file:
            if re.search(r'critical', line.lower()):
                print(line.strip())
```

extract\_critical\_errors('system\_logs.txt") to create a new system log file based on date and time stamp for each run.

Here's the modified script that creates a new system log file with a unique name based on the current date and time for each run. The script will extract critical errors from an input log file and write them to the newly created log file.

### Modified Script

```
import re
import datetime

def extract_critical_errors(input_log_file):
    # Generate a unique log file name based on date and time
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    output_log_file = f"critical_errors_{timestamp}.txt"

    # Open the input log file and create a new output log file
    with open(input_log_file, 'r') as infile, open(output_log_file, 'w') as outfile:
        for line in infile:
            if re.search(r'critical', line.lower()):  # Search for "critical" in each line
                outfile.write(line.strip() + '\n')  # Write the critical error to the new file

    print(f"Critical errors have been extracted to: {output_log_file}")

# Example usage
extract_critical_errors('system_logs.txt')
```

### How It Works

#### 1. Timestamp Generation:

- The `datetime.datetime.now()` function generates the current date and time.
- The `strftime()` method formats it into a readable string (e.g., `2025-04-04_21-30-15`).

#### 2. Dynamic Log File Creation:

- A new log file is created with a name like `critical_errors_2025-04-04_21-30-15.txt` for each run.

### 3. Critical Error Extraction:

- The script reads the input log file (`system_logs.txt`), searches for lines containing the word "critical" (case-insensitive), and writes those lines to the newly created log file.

### 4. Output Notification:

- A message is printed to indicate where the extracted errors have been saved.

## Example Output

If `system_logs.txt` contains:

```
INFO: System started successfully.  
CRITICAL: Disk space is running low.  
WARNING: High memory usage detected.  
CRITICAL: Network connection lost.
```

After running the script, a new file named `critical_errors_2025-04-04_21-30-15.txt` will be created with the following content:

```
CRITICAL: Disk space is running low.  
CRITICAL: Network connection lost.
```

## Benefits of This Modification

1. Ensures logs are organized by timestamp for easy tracking of runs.
2. Prevents overwriting previous logs by creating unique files for each execution.
3. Makes it easier to debug and archive extracted critical errors.

You can now use this script to dynamically generate logs based on your needs!