

Hier ging es um die die C++ Datei

Ich brauche ein C++ Script welches die Tastatur des Nutzers deaktiviert und ein Video abspielen lässt

Das, was du anfragst, klingt nach einem sehr invasiven Eingriff in das Betriebssystem, der Sicherheits- und Datenschutzrisiken birgt. Programme, die die Tastatur deaktivieren und Aktionen erzwingen, werden oft mit Malware in Verbindung gebracht. Ich werde den Code nur unter der Voraussetzung schreiben, dass es für einen legalen und ethisch korrekten Anwendungsfall ist, wie z. B. eine Präsentation, ein Kiosk-Modus oder Ähnliches.

Soll ich das Skript so gestalten, dass es:

1. Die Tastatur im Kiosk-Modus deaktiviert?
2. Ein Video im Vollbildmodus abspielt?
3. Eine Möglichkeit zum sicheren Beenden bietet?

Sobald ich das von dir bestätigt habe, starte ich mit dem Code.

Es ist für ein Schulprojekt wo es um IT-Security geht.
Das Video soll im vollbildmodus abgespielt werden.
Es soll eine sichere Methode zum beenden geben wenn man auf die "y" Taste drückt

Verstanden! Ich werde das Skript so gestalten, dass es:

1. Die Tastatur bis auf die Taste "y" deaktiviert.
2. Ein Video im Vollbildmodus abspielt.
3. Sicher beendet wird, sobald "y" gedrückt wird.

Ich erstelle jetzt das vollständige C++ Skript und strukturiere es sauber. Gib mir bitte einen Moment, um das auszuarbeiten. Ich melde mich, sobald es fertig ist.

`</>` Video Player With Keyboard Control

Das C++ Skript ist fertiggestellt und beinhaltet folgende Funktionen:

1. Es startet ein Video im Vollbildmodus (du musst den Pfad anpassen).
2. Die Tastatur wird überwacht, und das Programm beendet sich sicher, wenn "y" oder "Y" gedrückt wird.

Ich habe alles sauber strukturiert und kommentiert, damit du es leicht nachvollziehen kannst.

Soll ich noch einen Timeout oder eine zusätzliche Sicherheitsabfrage einbauen?

Welche Exceptions können auftauchen?
Wie könnte das Programm auf diese reagieren?

Gute Frage! Hier sind die potenziellen Fehlerquellen und wie das Programm darauf reagieren könnte:

Mögliche Exceptions:

1. Dateipfad nicht gefunden (Video-Datei)

- Das Video wird nicht abgespielt, weil der Pfad falsch oder die Datei nicht vorhanden ist.
- Lösung: Vor dem Start eine Überprüfung einbauen (`std::filesystem::exists`).

2. Fehler beim Starten des Videos

- Der `system()`-Aufruf kann fehlschlagen, z. B. bei fehlenden Rechten oder nicht unterstützten Dateiformaten.
- Lösung: Rückgabewert von `system()` auswerten und eine Fehlermeldung ausgeben.

3. Thread-Fehler beim Starten oder Beenden

- Wenn der Thread nicht korrekt gestartet oder beendet wird.
- Lösung: Fehlerauswertung mit `try-catch`.

4. Fehler beim Zugriff auf die Tastatur (selten, aber möglich)

- Wenn `_kbhit()` oder `_getch()` Probleme verursachen.
- Lösung: Fehlerbehandlung mit Exception Handling.

Ich kann das Skript für dich erweitern, um diese Fehler abzufangen und sauber zu handhaben. Soll ich das für dich ausarbeiten? Dann hast du ein vollständiges und robustes Beispiel für dein Schulprojekt.

Ja, bitte Bau es ein

Bearbeitet

Ich habe das Skript nun robuster gestaltet und folgende Verbesserungen vorgenommen:

Änderungen:

1. Überprüfung des Dateipfads:

- Vor dem Starten wird geprüft, ob die Videodatei existiert. Falls nicht, erscheint eine Fehlermeldung.

2. Fehlerbehandlung beim Video-Start:

- Der Rückgabewert von `system()` wird ausgewertet. Wenn das Video nicht startet, wird dies gemeldet.

3. Fehlerbehandlung bei der Tastaturüberwachung:

- Eventuelle Fehler bei `_kbhit()` und `_getch()` werden sicher abgefangen.

4. Thread-Sicherheit:

- Geprüft, ob der Thread korrekt gestartet und beendet wurde, bevor auf `.join()` zugegriffen wird.

Soll ich noch eine Art Logging einbauen, damit bei Fehlern ein Protokoll geschrieben wird? Oder reicht dir die Konsolenausgabe?

Ich will dass alle Meldungen in einer log Datei abgespeichert werden mit TimeStamp und anderen Dingen wo du meinst dass es wichtig ist

Bearbeitet

Ich habe das Programm jetzt um eine vollständige Logging-Funktionalität erweitert:

Neuerungen:

1. Log-Datei "application_log.txt"

- Alle wichtigen Ereignisse werden mit einem Zeitstempel protokolliert.
- Ereignisse beinhalten:
 - Programmstart
 - Start des Videos
 - Fehler beim Video-Start
 - Fehler beim Überwachen der Tastatur
 - Beenden durch "y"-Tastendruck
 - Programmende

2. Zeitstempel im Format `YYYY-MM-DD HH:MM:SS`

- Jede Log-Nachricht ist zeitlich genau nachvollziehbar.

Soll ich jetzt noch eine Art Fehlercode-Management einbauen, damit die Log-Datei noch spezifischere Fehlerinformationen enthält? Beispielsweise Exit-Codes für verschiedene Fehlerarten?

Welche Vorteile hat das Programm wenn man es einbaut?
Mach mir Vorschläge welche Funktionen noch eingebaut werden könnten

Vorteile durch Fehlercode-Management:

- **Bessere Fehleranalyse:** In der Log-Datei kann man genau sehen, warum etwas schiefgelaufen ist (z. B. Fehlercode 101 = Datei nicht gefunden, 102 = Berechtigungsproblem).
- **Schnellere Fehlerbehebung:** Durch klar definierte Fehlercodes kann man schneller herausfinden, wo der Fehler liegt.
- **Standardisierte Protokollierung:** Es ist einfacher, verschiedene Fehlerarten zu unterscheiden und gezielt zu beheben.

Vorschläge für weitere Funktionen:

1. Watchdog-Mechanismus:

- Überwacht das Programm im Hintergrund und startet es neu, falls es unerwartet abstürzt.

2. Video-Wiedergabe-Überwachung:

- Prüfen, ob das Video tatsächlich läuft, nicht nur der Prozess gestartet wurde.

3. Netzwerkprotokollierung (optional):

- Falls es für das Projekt relevant ist, könnten Netzwerkereignisse mitprotokolliert werden (z. B. wenn das Video aus einem Netzwerkpfad geladen wird).

4. Benachrichtigung bei Fehlern:

- Popup-Fenster oder ein akustisches Signal, falls ein Fehler auftritt.

5. Parameterübergabe beim Start:

- Möglichkeit, den Videopfad und andere Einstellungen direkt beim Programmstart als Parameter zu übergeben.

6. Log-Rotation:

- Die Log-Datei könnte nach einer bestimmten Größe automatisch archiviert werden, um Speicher zu sparen.

7. Sicherheitsabfrage vor dem Beenden:

- Ein kurzer Hinweis, wenn "y" gedrückt wurde: "Möchten Sie wirklich beenden? (j/n)".

Soll ich eine dieser Funktionen direkt einbauen? Ich würde vorschlagen, zuerst das Fehlercode-Management und die Sicherheitsabfrage. Das wäre ein guter nächster Schritt. Was meinst du?

Habe jetzt das Script ausgeführt und es funktioniert bei eingaben in die Konsole.
Wie kontrolliere ich die Eingabe auch außerhalb der Konsole?
Es geht darum weil sonst Schulkollegen trotzdem etwas eingeben

Sehr gute Frage! Momentan überwacht unser Programm nur Konsolen-Eingaben (`_kbhit()` und `_getch()`), aber nicht die globale Tastatur. Um auch globale Tastatureingaben abzufangen (auch außerhalb der Konsole), müssen wir auf Low-Level-Hooks der Windows-API zurückgreifen.

◆ Lösung: Globaler Keyboard Hook (Low-Level-Hook)

Damit können wir Tastenanschläge systemweit abfangen. Die wichtigsten Schritte sind:

1. Einen globalen Low-Level-Hook setzen.
2. Tastenanschläge filtern, außer die "y"-Taste.
3. Andere Tasten blockieren, solange das Video läuft.

Soll ich das direkt in den Code einbauen? Damit wäre sichergestellt, dass niemand während der Video-Wiedergabe etwas eingeben kann, egal wo sich der Fokus befindet.

Ich würde:

- Den globalen Hook integrieren.
- Die Erkennung der "y"-Taste direkt im Hook behandeln.
- Ein sauberes Beenden des Hooks beim Programmende sicherstellen.

Soll ich das so umsetzen?

Ja, bitte

Bearbeitet

Ich habe jetzt das Programm erweitert und folgende Änderungen vorgenommen:

◆ Globaler Low-Level Keyboard Hook:

- Die Überwachung der Tastatur ist jetzt global – unabhängig davon, ob der Fokus im Konsolenfenster liegt oder nicht.
- Alle Tasten außer "y" werden blockiert, solange das Video läuft.

◆ Erweiterungen im Code:

1. LowLevelKeyboardProc:

- Erkennt Tastendrücke global.
- Nur "y" wird akzeptiert; alle anderen Eingaben werden unterdrückt.

2. SetWindowsHookEx:

- Der Hook wird beim Start aktiviert und überwacht die Eingaben systemweit.

3. UnhookWindowsHookEx:

- Beim Beenden des Programms wird der Hook sauber entfernt.

Hinweis:

- Falls du das Programm testest, kannst du in anderen Anwendungen keine Eingaben machen, solange es läuft (außer "y").
- Drücke "y", um das Programm sauber zu beenden und die Kontrolle wiederherzustellen.

Soll ich noch eine Sicherheitsabfrage einbauen, wenn "y" gedrückt wird, damit man sich nicht aus Versehen beendet? Zum Beispiel: *"Möchten Sie wirklich beenden? (j/n)"*?

Füge hinzu dass auch mausinputs verarbeitet werden

Ich werde den Code jetzt erweitern, sodass zusätzlich zu den Tastatureingaben auch Maus-Eingaben global abgefangen und blockiert werden.

◆ Geplante Änderungen:

1. Einen Low-Level Mouse Hook hinzufügen.
2. Linksklick, Rechtsklick, Mausrad und Mausbewegungen blockieren.
3. Nur "y" auf der Tastatur beendet das Programm weiterhin.

Ich baue das jetzt direkt im Code ein. Gib mir einen Moment, ich melde mich gleich zurück!

</> Video Player With Keyboard Control

```
1  "#include <windows.h>  
2  #include <iostream>  
3  #include <thread>  
4  #include <atomic>  
5  
6  std::atomic<bool> running(true);  
7  HHOOK keyboardHook = NULL;  
8  HHOOK mouseHook = NULL;  
9  
10 // Low-Level Keyboard Hook Callback  
11 LRESULT CALLBACK LowLevelKeyboardProc(int nCode, WPARAM wParam, LPARAM lParam) {  
12     if (nCode == HC_ACTION) {  
13         KBDLLHOOKSTRUCT *pKeyboard = (KBDLLHOOKSTRUCT *)lParam;  
14
```

Bearbeitet

Stattdessen im Chat antworten >

Ich habe den Code jetzt erweitert:

◆ Neuerungen:

1. Globale Maus-Blockierung hinzugefügt:
 - Alle Mauseaktionen (Bewegungen, Klicks, Mausrad) werden abgefangen und blockiert.
 - Der Low-Level-Mouse-Hook sorgt dafür, dass keine Eingabe durchkommt.
2. Globale Tastatur-Blockierung bleibt:
 - Wie zuvor werden alle Tasten außer "y" unterdrückt.
3. Sauberes Beenden:
 - Drücken von "y" hebt den Hook auf und beendet das Programm.

Hier ging es um dass erstellen des provisorischen Video

Ich will einer mp4 Datei Untertitel und eine neue Tonspur geben
Ich habe subtitle edit und shotcut.
Kann ich mit beiden Programmen es bewerkstelligen?

Ja, mit Subtitle Edit und Shotcut kannst du dein Ziel erreichen: einer `.mp4`-Datei sowohl Untertitel als auch eine neue Tonspur hinzufügen. Hier ist, wie du das in Kombination mit beiden Programmen machen kannst:

✅ Schritt 1: Untertitel mit Subtitle Edit vorbereiten

1. Öffne die MP4-Datei in Subtitle Edit.
2. Erstelle oder bearbeite die Untertitel.
3. Speichere die Untertitel im `.srt`-Format (z.B. `film.srt`).

Hinweis: Subtitle Edit kann keine Untertitel direkt in die MP4-Datei einbetten – das macht später Shotcut.

✅ Schritt 2: Neue Tonspur mit Shotcut einfügen

1. Öffne die MP4-Datei in Shotcut.
2. Entferne oder stummschalte die bestehende Tonspur (falls nötig):
 - Rechtsklick auf die Tonspur im Zeitstrahl → „Stumm“ oder „Löschen“.
3. Importiere die neue Tonspur (z.B. eine WAV oder MP3-Datei).
4. Ziehe die neue Tonspur in eine Audio-Spur im Zeitstrahl.

✅ Schritt 3: Untertitel in Shotcut einfügen

Shotcut selbst unterstützt keine eingebetteten SRT-Untertitel beim Export. Es gibt zwei Optionen:

Option A: Untertitel hart einbrennen (Hardcoded)

1. Gehe auf `Filter` → `+` → `Text: Simple` oder `Text: Rich`.
2. Füge den Text manuell zu bestimmten Zeitpunkten hinzu (mühselig für ganze Filme).

Option B: Externe Untertiteldatei mitliefern

1. Exportiere die neue MP4-Datei mit der neuen Tonspur.
2. Benenne die `.srt`-Datei genauso wie die MP4-Datei (z.B. `film.mp4` und `film.srt`).
3. Stelle sicher, dass beides im selben Ordner liegt – viele Player (wie VLC) erkennen sie automatisch.

df

✅ Alternative für einfacheres Einbetten (optional)

Wenn du die Untertitel direkt weich (soft) oder hart (hard) einbetten willst, ist HandBrake ein besseres Tool als Shotcut:

- Öffnet MP4
- Fügt SRT-Dateien direkt hinzu
- Kann sie optional hart einbrennen oder soft beilegen

Fazit

Aufgabe	Subtitle Edit	Shotcut
Untertitel erstellen/bearbeiten	✅	❌
Tonspur ersetzen	❌	✅
Untertitel einbrennen	❌	⚠️ Nur manuell
Export mit neuer Tonspur	❌	✅

👉 Du kannst beide Programme nutzen, aber für eingebettete Untertitel wäre HandBrake einfacher, wenn du das willst.

Willst du, dass ich dir ein konkretes Beispiel (mit Export-Einstellungen) für Shotcut oder HandBrake gebe?

Wie exportiere ich das Video mit der neuen Tonspur?

Perfekt! Ich zeige dir Schritt für Schritt, wie du in Shotcut ein Video mit einer neuen Tonspur exportierst.

Schritt-für-Schritt: Neue Tonspur einfügen & Video exportieren in Shotcut

1. Video und neue Tonspur importieren

- Öffne Shotcut.
- Ziehe dein MP4-Video in das Vorschaufenster → klicke auf "Zur Wiedergabeliste hinzufügen".
- Ziehe das Video auf die Video-Zeitleiste (falls nicht automatisch dort).
- Importiere dann deine neue Tonspur (z.B. `.wav` oder `.mp3`) genauso.

2. Alte Tonspur entfernen (optional)

- Wenn du die Original-Tonspur entfernen willst:
 - Rechtsklick auf die Videospur im Zeitstrahl → "Audio trennen".
 - Die Audio-Spur erscheint separat – lösche oder stumm schalte sie.
 - Ziehe deine neue Tonspur in eine eigene Audiospur.

3. Tonspur synchronisieren

- Überprüfe, ob deine neue Tonspur synchron zum Video läuft.
- Passe sie ggf. im Zeitstrahl an.

4. Video exportieren mit neuer Tonspur

1. Gehe auf "Exportieren" oben rechts.
2. Wähle das Format aus – z.B. "Default" (MP4/H.264).
3. Optional: Unter "Erweiterte Optionen" → "Audio", stelle sicher:
 - Codec ist z.B. AAC oder MP3
 - Sample Rate auf 48000 Hz (Standard)
4. Klicke auf "Export-Datei".
5. Wähle Ziel und Dateinamen → Speichern.

Shotcut rendert jetzt dein Video mit der neuen Tonspur.

✅ Tipp: Schneller Test vor Export

- Spiele die Timeline in Shotcut ab, um sicherzustellen, dass nur die neue Tonspur zu hören ist.
- Exportiere erst, wenn alles gut synchronisiert ist.

Wenn du willst, kann ich dir auch ein Preset empfehlen oder helfen, wenn dein Ton leise oder asynchron wird. Sag einfach Bescheid!

Ich drücke auf Video Exportieren, aber es passiert nichts.
Woran könnte es liegen?

Wenn du in Shotcut auf „Video exportieren“ klickst und nichts passiert, kann das an ein paar häufigen Ursachen liegen. Hier sind die wahrscheinlichsten Gründe und Lösungen:

🔧 Mögliche Ursachen & Lösungen

1. ❌ Keine Datei ausgewählt oder keine Timeline

Problem: Du hast das Video nicht zur Timeline hinzugefügt.

✅ Lösung:

- Stelle sicher, dass dein Video im Zeitstrahl (Timeline) liegt.
- Wenn es nur in der Wiedergabeliste ist, passiert beim Exportieren nichts.
- Ziehe das Video in die Timeline, dann erneut „Exportieren“ → „Export-Datei“.

2. ❌ Du hast nur „Exportieren“ geklickt, aber nicht „Export-Datei“

Problem: Der Klick auf „Exportieren“ öffnet nur das Menü – es startet noch keinen Export.

✅ Lösung:

- Nach dem Klick auf „Exportieren“, musst du unten auf „Export-Datei“ klicken (grauer Button).
- Danach wählst du Speicherort & Name → Dann beginnt der Exportprozess.

3. ⚠️ Fehlender Zielpfad oder Schreibrechte

Problem: Shotcut kann die Datei nicht speichern, weil der Speicherort ungültig ist.

✅ Lösung:

- Wähle beim „Export-Datei“-Klick einen neuen Speicherort, z. B. den Desktop.
- Speichere unter einem einfachen Namen ohne Sonderzeichen (z. B. `video_neu.mp4`).

4. 🐛 Export hängt oder ist eingefroren

Problem: Shotcut startet den Export, aber zeigt nichts oder hängt.

✅ Lösung:

- Schau im „Jobs“-Fenster (rechts unten). Wird da ein Job angezeigt?
 - Grün = erfolgreich
 - Blau = läuft
 - Rot = fehlgeschlagen → Rechtsklick → „Protokoll anzeigen“ für Fehlermeldung
- Tipp: Falls Jobs-Fenster fehlt: Menü → „Ansicht“ → „Jobs“ aktivieren

5. 🌿 Fehlende Medien oder Formate

Problem: Manche Dateien (Audio, Video) fehlen oder werden nicht unterstützt.

✅ Lösung:

- Stelle sicher, dass alle Medien noch da sind (nicht verschoben).
- Vermeide exotische Audioformate – MP3 oder WAV sind ideal.

✅ Kurzcheckliste

Frage	Lösung
Video in Timeline?	Ja → Nur so wird exportiert
Hast du „Export-Datei“ geklickt?	Muss angeklickt werden nach dem Format auswählen
Speicherort korrekt?	Desktop wählen, einfacher Name verwenden
Jobs-Fenster zeigt Export?	„Jobs“ aktivieren und prüfen, ob ein Export läuft oder fehlgeschlagen ist
Problem weiterhin?	„Protokoll anzeigen“ im Jobs-Fenster für Fehlermeldung

Wenn du magst, kannst du mir auch einen Screenshot zeigen oder sagen, was im „Jobs“-Fenster steht – dann kann ich dir noch gezielter helfen.