

Table of contents

- Exception Handling
 - Einführung und Übersicht
 - Ablage der Protokolldatei
 - Aufbau der Protokoll-Datei
 - Diagnose

Exception Handling

Exception Handling

Einführung und Übersicht

ctrlX Applikationen besitzen einen Mechanismus, um Anwendungsausnahmen (Exception) zu protokollieren.

Die Protokollierung dient einer erweiterten Fehleranalyse von Anwendungen im Servicefall.

Tritt in einer ctrlX Applikation eine Anwendungsausnahme (Exception) auf, so wird diese durch das Laufzeitsystem abgefangen und in einer Protokolldatei erfasst (Exception-Log).

Anschließend wird das Auftreten der Anwendungsausnahme durch die integrierte LED-Statusanzeige signalisiert und die Anwendung beendet. Abhängig von der restart-condition der jeweiligen Anwendung wird diese ggf. automatisch neu gestartet.

Die Protokolldatei wird fortlaufend weitergeschrieben, wenn bei nachfolgenden Programmausführungen erneut Ausnahmen auftreten.

Hat die Protokolldatei eine Dateigröße erreicht, die mehr als 100 kByte beträgt, so wird eine Backup-Datei generiert und die Protokolldatei zurückgesetzt.

In der Protokolldatei werden folgende Informationen hinterlegt:

- Name des Anwendungsprogramms
- Datum und Uhrzeit im Format koordinierte Weltzeit (UTC – Universal Time, Coordinated)
- Signalnummer und Code
- Applikation-Register
- Aufrufhierarchie (Callchain) des Anwendungsprogramms

Ablage der Protokolldatei

Die Ablage der Protokolldatei erfolgt entsprechend der Snap-Namen unter folgendem Namen:

`var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log`

Das entsprechende Backup-File trägt zusätzlich die Erweiterung `.bak`:

`var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log.bak`

Beispiel für den Snap *rexroth-automationcore*

`/var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log`

```
/var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log.bak
```

Aufbau der Protokoll-Datei

Die Protokolldatei ist in die drei Sektionen unterteilt:

- "Application Exception"
- "Application Register"
- "Application Callchain"

Sektion "Application Exception"

Diese Sektion beinhaltet folgende Informationen:

- Name der Applikation
- Datum und Uhrzeit
- Signal-Nummer
- Bezeichner
- Signal-Code.

Weitere Informationen bezüglich der Signal-Nummern und Bezeichner sind im **"Portable Operating System Interface (POSIX), Standard IEEE 1003"** definiert.

Beispiel

```
----- APPLICATION EXCEPTION -----  
  
NAME: <APPLICATION NAME>  
DATE: 2020-03-10 05:47:23 GMT  
SIGNAL NUMBER: 11 (SIGSEGV)  
SIGNAL CODE: 128
```

Sektion "Application Register"

Diese Sektion beinhaltet die aktuellen, bei der Ausnahme aufgetretenen Pointer:

- "Instruction Pointer"
- Stack Pointer
- Base Pointer

Beispiel

----- APPLICATION REGISTER -----

Instruction Pointer (IP): 0x7ffefcf61d2

Stack Pointer (SP): 0x7ffbcfab910

Base Pointer (BP): 0x7ffbcfab950

Sektion "Application Callchain"

Die Sektion beinhaltet die Aufrufhierarchie in folgendem Anzeigeformat:

{Frame-Nummer} {Module-Name} ({optional Funktionsname} + {Modul-Offset} [{Funktionsadresse}]

Beispiel für Callchain mit fünf Frames

----- APPLICATION CALLCHAIN -----

0 /snap/foo/x0/foo/.cache/bundle1/version0.0/libfoo.so(bundleActivator_stop+0x36)
[0x7ffefcf61d2]

1 /lib/x86_64-linux-gnu/libpthread.so.0(+0x76db) [0x7fff653f6db]

2 /lib/x86_64-linux-gnu/libc.so.6(clone+0x3f) [0x7fff712a88f]

3 /lib/x86_64-linux-gnu/libc.so.6(start+0x3f) [0x7fff712345f]

4 /lib/x86_64-linux-gnu/libc.so.6(init+0x113f) [0x7fff79aaddf]

Diagnose

Das Auftreten einer Ausnahme wird nach dem Erstellen der Protokolldatei an der integrierten Status-LED-Anzeige der ctrlX über den Betriebszustand "System Fehler" - SysError signalisiert.

Die LED-Zustände für ctrlX CORE sind in [„LED-Diagnosecode“](#) beschrieben.

Einführung und Übersicht

Einführung und Übersicht

ctrlX Applikationen besitzen einen Mechanismus, um Anwendungsausnahmen (Exception) zu protokollieren.

Die Protokollierung dient einer erweiterten Fehleranalyse von Anwendungen im Servicefall.

Tritt in einer ctrlX Applikation eine Anwendungsausnahme (Exception) auf, so wird diese durch das Laufzeitsystem abgefangen und in einer Protokolldatei erfasst (Exception-Log).

Anschließend wird das Auftreten der Anwendungsausnahme durch die integrierte LED-Statusanzeige signalisiert und die Anwendung beendet. Abhängig von der restart-condition der jeweiligen Anwendung wird diese ggf. automatisch neu gestartet.

Die Protokolldatei wird fortlaufend weitergeschrieben, wenn bei nachfolgenden Programmausführungen erneut Ausnahmen auftreten.

Hat die Protokolldatei eine Dateigröße erreicht, die mehr als 100 kByte beträgt, so wird eine Backup-Datei generiert und die Protokolldatei zurückgesetzt.

In der Protokolldatei werden folgende Informationen hinterlegt:

- Name des Anwendungsprogramms
- Datum und Uhrzeit im Format koordinierte Weltzeit (UTC – Universal Time, Coordinated)
- Signalnummer und Code
- Applikation-Register
- Aufrufhierarchie (Callchain) des Anwendungsprogramms

Ablage der Protokolldatei

Ablage der Protokolldatei

Die Ablage der Protokolldatei erfolgt entsprechend der Snap-Namen unter folgendem Namen:

`var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log`

Das entsprechende Backup-File trägt zusätzlich die Erweiterung `.bak`:

`var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log.bak`

Beispiel für den Snap *rexroth-automationcore*

`/var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log`

`/var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log.bak`

Aufbau der Protokoll-Datei

Aufbau der Protokoll-Datei

Die Protokolldatei ist in die drei Sektionen unterteilt:

- "Application Exception"
- "Application Register"
- "Application Callchain"

Sektion "Application Exception"

Diese Sektion beinhaltet folgende Informationen:

- Name der Applikation
- Datum und Uhrzeit
- Signal-Nummer
- Bezeichner

- Signal-Code.

Weitere Informationen bezüglich der Signal-Nummern und Bezeichner sind im "Portable Operating System Interface (POSIX), Standard IEEE 1003" definiert.

Beispiel

```
----- APPLICATION EXCEPTION -----  
  
NAME: <APPLICATION NAME>  
DATE: 2020-03-10 05:47:23 GMT  
SIGNAL NUMBER: 11 (SIGSEGV)  
SIGNAL CODE: 128
```

Sektion "Application Register"

Diese Sektion beinhaltet die aktuellen, bei der Ausnahme aufgetretenen Pointer:

- "Instruction Pointer"
- Stack Pointer
- Base Pointer

Beispiel

```
----- APPLICATION REGISTER -----  
  
Instruction Pointer (IP): 0x7ffefcf61d2  
Stack Pointer (SP): 0x7ffbcfab910  
Base Pointer (BP): 0x7ffbcfab950
```

Sektion "Application Callchain"

Die Sektion beinhaltet die Aufrufhierarchie in folgendem Anzeigeformat:

{Frame-Nummer} {Module-Name} ({optional Funktionsname} + {Modul-Offset} [{Funktionsadresse}]

Beispiel für Callchain mit fünf Frames

```
----- APPLICATION CALLCHAIN -----  
0 /snap/foo/x0/foo/.cache/bundle1/version0.0/libfoo.so(bundleActivator_stop+0x36)  
[0x7ffefcf61d2]  
1 /lib/x86_64-linux-gnu/libpthread.so.0(+0x76db) [0x7fff653f6db]  
2 /lib/x86_64-linux-gnu/libc.so.6(clone+0x3f) [0x7fff712a88f]  
3 /lib/x86_64-linux-gnu/libc.so.6(start+0x3f) [0x7fff712345f]  
4 /lib/x86_64-linux-gnu/libc.so.6(init+0x113f) [0x7fff79aaddf]
```

Diagnose

Diagnose

Das Auftreten einer Ausnahme wird nach dem Erstellen der Protokolldatei an der integrierten Status-LED-Anzeige der ctrlX über den Betriebszustand "System Fehler" - SysError signalisiert.

Die LED-Zustände für ctrlX CORE sind in [↘ „LED-Diagnosecode“](#) beschrieben.