Der IOCCC
Wie man mit schlechtem Code gewinnt

Volker Diels-Grabsch

14. Januar 2020

Übersicht

Übersicht

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Geschichte

- seit 1984
- einer der ältesten Wettbewerbe im Netz
- Usenet: net.lang.c
- Jury damals: Landon Curt Noll, Larry Bassel
- heute: Landon Curt Noll, Leonid A. Broukhis, Simon Cooper
- 26. Runde: 2019

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Ziele

- Unverständlichkeit von C-Programmen auf die Spitze treiben
- Trotzdem Regeln beachten!
- Demonstrieren, wie wichtig guter Programmierstil ist.
- C-Compiler stressen!
- Die subtilen Ecken von C illustrieren
- Ein sicheres Forum für schlechten C-Code.
- $\Rightarrow$ Müllhalde für C-Code? Nein!

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - Tehenrad
  - pearlabel
  - möglichst keine Compiler-Warnungen: -Werror -W -Wall -Wextra -Weverything -Wbad-taste -std=c99
  - eigene Subkultur
  - Viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
    - Verkomplizierte „Hello World!“ haben es schwer
    - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
    - fehlerfrei
    - portabel:
    - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
    - eigene Subkultur
    - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (altered) nagelneue Compiler
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (altere/neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (altered-nature Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (altere/neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Qualitäten

- Verschleierungstechniken sind eine Kunst
- Unterliegen Mode-Erscheinungen
- Programme sollen Sinnvolles oder Interessantes tun
  - Verkomplizierte „Hello World!“ haben es schwer
  - Intelligente Größenbeschränkung schützt vor Nadel-Heuhaufen-Taktik
  - fehlerfrei
  - portabel:
    - mindestens GCC + Clang
    - mindestens 32 + 64 Bit
    - (ältere+neuere Compiler)
  - möglichst keine Compiler-Warnungen: -Werror -Wall -Wextra -Weverything -pedantic -std=c99
  - eigene Subkultur
  - viele Insider-Scherze

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - loccsize $\leq$ 2057
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein $\neg$M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - Locsize $\leq$ 2057
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - Loccount $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

Regeln

- 26 Regeln
- viele selbstironisch
- einige Formalitäten
- einige wichtig:
  - Programmgröße $\leq$ 4096 Bytes
  - iocccsize -i $\leq$ 2053
  - keine vorhandenen Dateien modifizieren
  - harte Deadline
  - Einreichungen pro Runde $\leq$ 8,000000
  - zuverlässiger, automatischer Build
  - kein 8-bit-Zeichen, kein ^M
  - Zusatzdateien erlaubt, aber kein Code und $\leq$ 1 MiB

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

- Regeln entwickeln sich.
- Kreative Umgehung von Regeln wird belohnt.
- Aber Regeln werden verschärft.
  - Beispiel: Warum wird heute Portabilität verlangt?
- ergänzt durch Richtlinien
- ergänzt durch Hinweise in Jury-Bemerkungen
- ergänzt durch Hinweise, was für Einträge es knapp nicht schafften

Übersicht

Übersicht

Ablauf

- 27. Runde beginnt morgen!
  - 2020-Jan-15 15:15:15 UTC
- Macht mit!
- https://www.ioccc.org/

Ablauf

- 27. Runde beginnt morgen!
  - 2020-Jan-15 15:15:15 UTC
- Macht mit!
- https://www.ioccc.org/

Ablauf

- 27. Runde beginnt morgen!
  - 2020-Jan-15 15:15:15 UTC
- Macht mit!
- https://www.ioccc.org/

Ablauf

- 27. Runde beginnt morgen!
  - 2020-Jan-15 15:15:15 UTC
- Macht mit!
- `https://www.ioccc.org/`

Der IOCCC
Wie man mit schlechtem Code gewinnt

Volker Diels-Grabsch

https://njh.eu/

CC BY SA