**Grundlagenpraktikum Rechnerarchitektur**

1. **Woche**

* Einführung x86-64:
* Eine Adresse ist 64 Bit lang, so sind die Register
* Die Operationen mit Immediate gehen nur mit mov

Table

Description automatically generated

* Intel SDM
* Es gibt solche Instruktionen wie CMOVL EAX, ECX: EAX wird auf ECX gesetzt wenn Status-Flag 1 anzeigt, und 32 oberen Bit von RAX werden auf 0 gesetzt.
* POPCNT RAX, RCX zählt gesetzte Bits in RCX und schreibt das Ergebnis in RAX.
* CQO setzt 32 oberen Bits in RAX auf 1 in dem Fall von Sign-Extension.
* Betriebssystemenschnittstelle
* Ein Rückgabewert in Range von -1 bis -4095 indiziert einen Error.
* Ein Programm wird mit execve durchgeführt und es gibt keinen Rückgabewert wenn erfolgreich.
* Daten zu Programmstart (auf Stack)

1. Programmargumente
2. Umgebungsvariable
3. Auxilliary Vector

z.B. $ ./testprog arg1 arg2

Table

Description automatically generated

* Während der Benutzung der Adresse in der Operation, muss man *ptr* hinzufügen.
* Lokale Label innerhalb einer Funktion mit .L, Kommentare kann man mit // oder # einleiten, Blockkommentare mit /\*\*/.

1. **Woche**

* Einführung C (1970er)

Table

Description automatically generated

* Die Funktion void foo() nimmt beliebig viele Parameter. Really? Man sollte void
* foo(void) deklarieren, wenn man die Funktion ohne Parameter anrufen will.
* J hat den Wert -34 nach dem Statement int j = -042 😊, why?

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence

* Debugging mit GDB:
* Kompiliere das Programm mit -g Flag.
* Dann ruf gdb mit dem Syntax gdb [program\_name], b für breakpoint, r für run, s für step
* Benutze *layout prev* um in Assemble mode zu wechseln.
* Für Anzeigen von XMM Registern, benutze **p /x $xmm0.v2\_double** für den double typ
* Benutze print um die Werte von Register zu sehen: z.B. print len;
* Für pointers, wenn wir print verwenden, können wir nur die Adresse sehen, wenn wir die Werte sehen wollen, müssen wir x benutzen: z.B. x factors zeigt den ersten Wert von factors (factors ist ein Array)
* x/3a argv: print 3 Adressen von argv, respectively x/s = string, x/d = decimal
* Man kann s für step benutzen oder si für step instruction benuzten, um jede Assemble Instruktion zu sehen.
* Quit verlässt gdb
* Info break lists all the break points
* Delete/disable a breakpoint

1. **Woche**

* Datenstruktur in C
* Sizeof() gibt die Größe von einer Variable in Bytes zurück.
* Pointer:

A picture containing chart

Description automatically generated

A picture containing text

Description automatically generated

* Die Pointers sind nebeneinander gespeichert, so die Subtraktion ergibt sich den Wert 1. Die Addition von 2 Pointers geht’s nicht.
* 2 Möglichkeiten für Zugriff auf ein Pointer: \*(ptr+5) und ptr[5]
* Void Pointer: A void pointer is a pointer that has no associated data type with it. A void pointer can hold address of any type and can be typecasted to any type.
* Ein int-Pointer kann in einen char Pointer gecasted werden. Aber umgekehrt geht es nicht, da int strengere Alignmentanforderungen stellen als char.

Text

Description automatically generated

* Eine Dereferenzierung eines void-Pointers ist nicht möglich!

Graphical user interface, text, application, email

Description automatically generated

* Konversionen von und zu void -Pointer sind kein Problem

Text

Description automatically generated

* Nicht explizit definierte Elemente werden zu Null initialisiert

Chart, waterfall chart

Description automatically generated

* Unterschied zwischen Array und Pointer

Chart

Description automatically generated with low confidence

Text

Description automatically generated with medium confidence

* Struct:
* Zusammengesetzter Datentyp mit Elemente verschiedener Typs, Elemente liegen aufeinanderfolgend im Speicher.

Graphical user interface

Description automatically generated with medium confidence

* Initialisierung:

Text

Description automatically generated

* Der Wert von size1 wird 8 sein, der von size2 aber 12! Das liegt daran, dass nicht nur die Folge-Adresse des struct die Alignment-Anforderungen des struct erfüllen muss, sondern dass auch für die einzelnen Elemente innerhalb des struct ihre eigenen Alignment-Anforderungen erfüllt werden müssen. D.h. für Penguin2, dass id 4-Byte aligned sein muss und nach age ein 3-Byte-Padding erfolgen muss. Dann muss aufgrund von color noch einmal ein 3-Byte-Padding eingefügt werden. Insgesamt führt das zu 1 + 3 + 4 + 1 + 3 = 12 Byte Größe für Penguin2. Bei Penguin1 müssen hingegen lediglich zwei Byte Padding eingefügt werden, da für id das Alignment bereits korrekt ist und somit die Größe des structs nur noch auf ein Vielfaches von 4 erweitert werden muss.

Chart

Description automatically generated

* Union

Graphical user interface, application, Word

Description automatically generated

* In 2-dimensional Array kann obere Größe fehlen, z.B. matr[][4];
* Speicherbereich
* Inhalte von Speicherbereiche

Text, table

Description automatically generated with medium confidence

* Alloca()

Graphical user interface, text, application

Description automatically generated

* Fflush()

A picture containing text

Description automatically generated

1. **Woche**

* Sichere Programmierung
* Alle Overflows basieren auf dem Nullterminal. "erap" enthält implizit eines, weshalb b nicht genug Platz für den String bietet. scanf hängt noch ein Nullterminal an den gelesenen String an. Der Buffer d ist nur ein Byte groß, "a" enthält aber auch wieder ein Nullterminal.

Chart

Description automatically generated

* Sichere Funktionen: memcpy(), fgets(), strncpy(), strncat()
* Unsichere Funktionen: gets(), scanf(), strcpy(), strcat()
* Antwort 4: Da der Inhalt von buf aufgrund des Einlesens mit scanf durch den Nutzer frei wählbar ist, kann dieser dort auch Format Specifier wie %s, %d, etc. angeben. printf wird diese Format Specifier dann wie üblich interpretieren und als Konsequenz evtl. den Inhalt von Registern- und/oder des Stack-Speichers ausgeben

A picture containing graphical user interface

Description automatically generated

* Vermeiden von memory leak

Text

Description automatically generated

* Programmargumente in C
* Wenn getopt auf eine Option trifft, die nicht im optstring spezifiziert wurde, wird der character '?' zurückgegeben. Dieser und alle anderen Rückgaben außer 'n' und 't' führen in den default case und damit zum Ausdrucken einer Fehlernachricht und vorzeitigem Beenden mit einem Fehlercode.

Text

Description automatically generated

* System V ABI (Application Binary Interface):
* Datentypen (Größe, Alignment,…)
* Calling Convention (Ort der Funktionsparameter, Rückgabewert, welche Register zu sichern)

Text, table

Description automatically generated with medium confidence

* Speicherstruktur
* Betriebsystemschnittstelle
* fun1 speichert das Ergebnis der Division in r10, dann wird fun0 aufgerufen. Da r10 ein temporäres Register ist, kann es passieren, dass fun0 den Wert in r10 ändert. Nach Rückkehr wird dann der Wert in r10 zurückgegeben. Zudem wird hier das 16-Byte-Alignment zerstört, da call 8 Byte auf den Stack legt und davor keine Korrektur des Stackpointers oder ein weiterer push stattgefunden hat

A picture containing graphical user interface

Description automatically generated

* Struct als Parameter

Text

Description automatically generatedText, letter

Description automatically generated

Graphical user interface, text

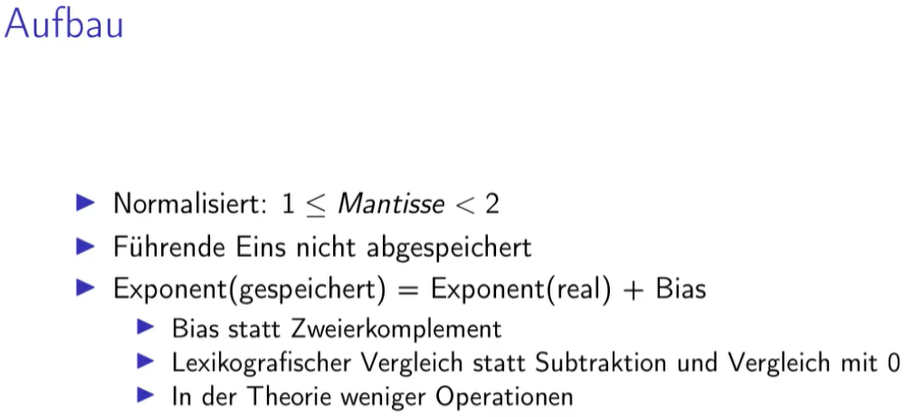
Description automatically generated

1. **Woche**

* Floating Point
* Floating Point Struktur

A picture containing chart

Description automatically generated



* Double

Table

Description automatically generated

* Unendlich/ Infinity: Infinity ist eine Zahl und – Infinity < + Infinity

A picture containing table

Description automatically generated

* Not a number: Gleichheit mit NaN sind immer false.

Table

Description automatically generated with low confidence

* Andere Floating Point Formate

Table

Description automatically generated

* SEE
* SEE Register

Application

Description automatically generated with low confidence

* Instruktionen: pxor dst, src; movss dst, src; movd/movq (move from general purpos registers)
* Addss: addition 2 unteren 32 bit von 2 xmm Registern (analog für subss)
* Divss, mulss: division, multiplikation 2 floating point Nummer
* Cvtsi2ss: konvertiert eine Ganzzahl in einen Single-Precision (analog zu cvtss2si)
* Namenskonvention
* Ss: Scalar Single
* Sd: scalar double
* Vergleiche:
* Ucomiss (vorzeichenlos) (ucomisd für double)

Text

Description automatically generated with medium confidence

* Calling Convention

Text

Description automatically generated

A picture containing text

Description automatically generated

1. **Woche**

* SIMD mit SSE
* Arbeitung auf ganzen XMM-Register zur gleichen Zeit
* Keine Speicher zu Speicher Operationen, mindesten ein XMM-Register
* Kenntwort für SIMD (P: Packed) in der Operation
* Die Teile arbeiten unabhängig voneinander
* Ideal für SIMD: wenn Daten nebeneinander im Speicher liegen und fortlaufen und das XMM-Register mit möglichst wenig Speicherzugriff gefüllt wird.
* Instruktionen:
* PADDD: Addiert 4 DWORD Integer auf 4 andere DWORD Integer (analog: PADDB, PADDQ)
* MOVD: kopiert ein 32 bit Nummer in XMM0
* PSHUFD xmm0, xmm0, 0x00: kopiert den niedersten 32 bit Wert in drei (0x00 is Shuffle-Maske) höherwertigen DWORD in XMM0
* MOVDQU xmm1, [rdi]: lädt 4 Integer aus dem Speicher ab RDI in XMM1
* MOVDQU [rdi], xmm1: schreibt 4 Integer aus XMM1 in Speicher ab RDI

Graphical user interface, text

Description automatically generated

* Gleitkomma Instruktionen:
* ADDSS: addiert einen Float auf einen anderen Float (Skalar Single-Precision Floating Point) => keine SIMD Befehl
* ADDPS: addiert 4 Float auf 4 andere Float
* ADDPD: addiert 2 Double auf 2 andere Double
* MOVUPS xmm0, [rdi]: lädt 4 Floats ab der Speicheradresse RDI in XMM0

Text

Description automatically generated

* Die meisten SIMD-Instruktionen erfordern **16Byte Alignment**
* Ausnahme von 16Byte Alignment: **Skalar SSE-Instruktionen** wie ADDSS
* Aligned Zugriff:
* MOVAPS xmm/m128, xmm/m128 (Move Aligned Packed Single-Precision): Speicher-Operanden müssen aligned sein. (MOVUPS erfordert Alignment nicht)
* SIMD Stolperfallen
* Höhere Leistungsaufnahme/Temperatur mit möglicher Taktreduzierung
* SIMD mit C
* Neuer Datentyp: \_\_m128 (es wird nicht explicit ein Register zugeordnet)
* Instruktionen
* \_mm\_add\_ps (\_\_m128 a, \_\_mm128 b) (ähnlich wie addps in assembly)
* \_mm\_mul\_ss (\_\_m128 a, \_\_mm128 b) (ähnlich wie mulss in assembly)
* \_mm\_load\_ps(float const\* mem\_addr) (movaps und movups): Inhalt von Speicher ab mem\_addr wird in ein \_\_m128 geschrieben.
* \_mm\_store\_ps(float const\* mem\_addr, \_\_m128 a) : Inhalt von a wird in mem\_addr geschrieben
* \_mm\_load\_ss: ähnlich wie \_mm\_load\_ps aber nur ein Float in niedrigwertigen 32bit.

Text, letter

Description automatically generated

* Andere Datentypen
* \_\_m128d: für double (mit Instruktion wie \_mm\_add\_pd)
* \_\_m128i: für Integer (mit Instruktion wie \_mm\_add\_epi8)
* Flags:
* GCC automatisch ab -O3 (vektorisieren)
* -march=native
* -fopt-info-vec(-missed) gibt aus ob Vektorisierung erfolgreich war.
* objdump
* Optimierung in GNU C
* Compiler-Optimierungen

Table

Description automatically generated with medium confidence

* Optimierung von Berechnungen
* Constant Folding: Konstante bei Kompiler Zeit berechnen
* Constant Propagation: Variablen werden mit ihren Wert ersetzt
* Common Subexpression Elimination:
* Optimierung von Funktionsaurufen: Tail Call Optimization

Graphical user interface, text, application

Description automatically generated

* Low-level Optimierungen:
* Instruktion Selection: ersetzen Multiplikation durch SHL
* Register Allocation: vermeiden von Stack zugriff
* Layout von Datenstrukturen: Structs

Text, letter

Description automatically generated

1. **Woche**

* Weitere SIMD konzept
* Vergleich 2 XMM-Register: PCMPEQB (P: packed, CMP: compare, EQ: equal, B:byte) (analog für PCMPEQW, PCMPEQD, PCMPEQQ, PCMPGTW,…)

Table

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

* AVX Register
* Datenbreite: 256 Bit, ymm0 bis ymm15, drei Operanden Format

Graphical user interface, text

Description automatically generated with medium confidence

* Instruktionen:
* VADDPS

Chart

Description automatically generated with medium confidence

* VMOVSD



* VBROADCASTSS xmm/ymm, xmm/m32

Diagram

Description automatically generated with low confidence

Text, letter

Description automatically generated

Graphical user interface, application, table

Description automatically generated

* Benchmarking
* Zeitmessung mit *time:* time ./main
* Nachteil: Alles wird gemessen, auch Eingabe und printf,…
* Zeimessung im Code:
* Abstand zwischen Messpunkten mindesten eine Sekunde: genügend Workload durch zB Schleife
* Umgebungsbedingungen
* Überprüfung von Korrektheit
* Die Messung mindesten 3 mal wiederholen
* Dokumentation
* Reproduzierbarkeit der Ergebnisse
* Setup des System (CPU, Speicher, Betriebsystem)
* Kompiliervorgang (Compiler, Version, Optionen)
* Testvorgang (Eingaben, Wiederholungen)
* Aufbereitung von Ergebnisse in Diagramm

Chart, line chart

Description automatically generated

Text, application

Description automatically generated

1. **Woche**

* Perf
* Perf record ./main (prüft die Instruktionen und ihre Durchführungszeit)
* Perf record (liest die Rekorde)
* Perf list (gliedern alle Funktionalitäten mit Perf)
* Cache Misses: Perf record -e cache-misses ./main
* Git basics

Diagram

Description automatically generated

* Git branch -l (gliedern alle vorhandene branches)
* Git checkout -b testing (wechseln in testing branche)
* Git push -u origin testing
* Git merge testing (merge mit dem Branche Testing)
* Git push (löschen die anderen Branches)

Hausaufgaben

* 1.Woche
* Gauß:

A picture containing graphical user interface

Description automatically generated

Lea: Berechnet den Wert von der Adresse, in dem oberen Fall: lea rax, [rdi +1] = mov rax, rdi; inc rax;

* Fakultät:

Graphical user interface, application, Word

Description automatically generated

* 2.Woche
* Collatz()

Text

Description automatically generated

* EAN13()

Text

Description automatically generated

* 3.Woche
* Listsum()

Text

Description automatically generated

* Brainfuck()

A picture containing text

Description automatically generated

* 4.Woche
* Memccpy

Graphical user interface, text, application

Description automatically generated

* Map()

Graphical user interface

Description automatically generated with medium confidence

* 5.Woche
* Circ()

Graphical user interface, text, application

Description automatically generated

* Numquad()

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

* Round:

Text

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

* 6.Woche
* SAXPY

Graphical user interface, text, application

Description automatically generated

* SDOT

Graphical user interface, text

Description automatically generated

* 7.Woche
* 8.Woche

**Quiz**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Quiz: Woche 2

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Zusatzaufgabe:

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface

Description automatically generatedTable

Description automatically generatedA picture containing graphical user interface

Description automatically generated