

Projektlogg- Malin Englund

2/12 Möte med Stephan- ½ timme ish

Vi kom fram till att första steget var att dela upp allt i moduler och sen sätta igång med kodningen.

4/12 Gruppmöte- 3 tim

Vi gjorde ett gemensamt försök att förstå projektspecifikationen och delade upp de svårförstådda delarna till olika gruppmedlemmar för att läsa på. Jag fick unioner.

6/12 Läsa på om ämne- 4 tim

Jag upptäcker att unioner inte är obligatoriskt så jag läser på om plattforms skillnader istället. Jag och Haubir läser på om endianness.

8/12 Gruppmöte- 3 tim

Alla ska ha läst på om olika saker. Bartlett algoritmen blir en stor diskussion och alla är super förvirrade. Alla berättar lite om sin sak.

9/12 Implementering – ½ tim

Jag skrev ihop lite kod om endianness.

10/12 Gruppmöte- 2 tim

Vi delar upp moduler. heap.c, collector.c och traverser.c.

installering av cunit 3 tim

Installera cunit och fick det tillslut att funka.

14/12 försök till implementering/planering 1 tim

försökte göra collector.c men insåg att vi inte kunde göra någonting innan de andra modulerna var klara.

15/12 Möte med stephan- 1 tim

Stephan sa att vi hade en bra start till moduluppdelning men att vi skulle behöva dela upp mer. traverser.c blev heap_traverser.c och stack_traverser.c. Jag och Haubir fick traversera stacken.

Implementering – 4 tim

Jag och Haubir försöker traversera och printa stacken.

16/12 Implementering- 4 tim

Jag och Haubir fortsätter med våra försök att traversera stacken. Vi börjar på även på listan.

20/12 Implementering – 1 tim

Jag sitter själv och pillar och lyckas köra koden på flera olika plattformar.

21/12 Implementering – 5 tim

Jag och Haubir fortsätter med stacktraverseringen. Vi har skrivit vidare på listan så vi traverserar stacken åt olika håll beroende på åt vilket håll den växer. Vi använder Carl och Andres `validate_object` funktion(ej implementerad än dock utan returnerar alltid true) för att kolla om pekarna på stacken är "levande" pekare. Vi bör fortfarande kolla vidare på detta då vi inte riktigt fattar hur det ska gå till och eventuellt att det behövs någon mer koll.

23/12 Implementering – 4 tim

Vi vet hur vi ska kolla om en pekare pekar på vår heap, men koden i `heap.c` är knasig så vår kod funkar inte just nu. Vi har skrivit en funktion för att kolla åt vilket håll heapen växer. Koden tycks funka på fries och yxan men mer testning behövs. På tussilago knasar kompileringen och vi fattar inte varför.

27/12 Testning 3,5 tim

Vi har läst på om enhetstestning. Målet för idag var att börja skriva enhetstester för `stack_traverser.c`. Haubir skrev en h-fil för `stack_traverser.c`. Det tog ganska lång tid att fatta hur man skulle göra och det finns mycket kvar. Jag lyckades få ihop lite tester för `get_stack_top` och `stack_grows_from_top`. Haubir koncentrerade sig mer på suites och trolleri i makefilen. Vårt mål är att bli klara med testerna tills den 29/12.

28/12 3 tim

Läste på om cunit.

29/12 Testning 2 tim

Skriver klart de flesta tester med Haubir. Jag får i uppgift att fixa makefilen då det var något kompileringsfel och Haubir skriver vidare på testerna.

31/12 Implementering 30 min

Försöker få makefilen att funka men lägger av då Carl fixar den innan mig.

2/1 Testning 3 tim

Det jag skulle göra idag var att fixa så att fler tester går igenom. Det tog sjukt lång tid att ens köra koden, makefilen var tydligen inte 100%igt fixad. Det är något knasigt kompileringsfel på tussilago och yxan, så försökte ge mig på fries istället. Lyckades inte lösa någon assertionsfel men fick i alla fall lite insyn i testfallen.

3/1 Implementering 2 tim

Fråga Carl om `is_pointing_heap` funktionen. Något tycks var fel med stackpointer-avrefereringen. Senare upptäckte Haubir att det snarare har att göra med endaddressen från `heap.c`.

5/1 5 tim Dokumentering testning

Jag kommentera `stack_traverser.h`. Fortsatte med makefil trasslet.

6/1 Implementering 5 tim

Makefilen är efter idag felfri. Körde på olika plattformar. Upptäckte att `std=c11` inte funkar på tussilago så körde `c99` istället. Fick då kompileringsfel istället så försökte lösa det utan framgång.

7/1 Testning, implementering 6 tim

Jag har försökt fixa till assertionsfelen i testfilen, då det troligtvis är fel på testen och inte koden. Råkade dock skapa segfaults istället för att jag inte vet hur man använder strcmp, vilket tog ett tag att fundera ut med hjälp av gdb. Vi gjorde top till en dubbelpekare men tog bort det för att koden knasa ur. Jag fixade i is_pointing_at_heap men den funkar dock fortfarande inte men den gör i alla fall mindre fel. Vi fixade till stack_grows_from_top och ändrat det obegripliga namnet. Vi tog bort heap_grows_upwards eftersom den är onödig.

8/1 Testning 2 tim

Jag försöker fixa testet för get_alive_stack_pointers.

9/1 Testing 4 tim

fortsätter med att fixa testerna men ingenting funkar!!!

10/1 Implementering/testning 12 tim

Vi trasslade runt med dubbelpekare och vi löste till slut problemet med att vi la till stackadressen istället för heapadressen. Jag skrev tester för is_pointing_at_heap och get_alive_stack_pointers.

11/1 Postmortem, dokumentering 12 tim

Ingenting kompilera till att börja med så det tog lite tid att fixa. Jag skrev reflektionerna, kommenterade kod och började med inlupp2 grejen som ska skräpsamlaren ska testas på.

12/1

Skickade inlupp2 filerna till Haubir(jag hade tenta).