Namn: Haubir Mariwani IT2
Grupp: Fladdermusen

2/12 Möte, Planering
Jag missade det första projektmötet men vi kom överens om en ny tid vi ska träffas på och planera hur vi ska göra detta.
Tid nedlagd: 0h

4/12 Möte, Planering
Gruppmöte 1311
Vi satte oss ner idag för att försöka förstå specen helt och hållet tillsammans. Resultatet blev att vi delade upp det hela i kunskapsområden som var och en ska kolla upp. Min uppgift blev att undersöka hur man gör programmet ska bli plattformsoberoende.
Tid nedlagd: 3h

6/12 Planering
Då Malins sak att kolla upp anseddes överflödig för detta projekt (unions) så läste jag och hon tillsammans på om endianness, little och big, och forskade kring skillnader mellan olika plattformar och hur man jobbar runt dem.
Tid nedlagd: 6h

8/12 Möte, Planering
Gruppmöte
Alla hade läst på om olika saker och berättade om vad som behöver göras. Blev förvirrning kring Bartletts algoritm så det gicks igenom noga tillsammans så att man skulle förstå den så bra som möjligt. Tyvärr har jag fått dåliga nyheter om en nära vän vilket påverkat mig mentalt idag så jag kommer behöva åka hem till Stockholm och vara med familjen för att klara av detta. Väldigt mentalt påfrestande. Har berättat för gruppen hur det ligger till och de förstod.
Tid nedlagd: 3h

10/12 Möte, Planering
Gruppmöte, moduluppdelning.
Gruppen har haft möte idag men jag kunde inte medverka då jag fortfarande är hemma och bearbetar personliga saker. De har delat upp sig i programmeringspar och Malin hamnade ensam så jag tänker bilda par med henne om några dagar. Malin hade fått garbage-collector biten att skriva på.
Tid nedlagd: 0h

14/12 Implementering
Vi märkte rätt snabbt att det inte fanns mycket att göra då denna modul först kan skrivas då de andra modulerna är klara. Men vi gjorde ändå ett försök i att lista ut vad vi kunde göra med det lilla vi hade. Gick inte bra. Vi är förvirrade kring denna modul och tänker ta upp det på morgondagens möte.
Tid nedlagd: 2h

15/12 Möte, Implementering, Planering
Möte och Planering med Stephan

Stephan berättade att det var onödigt att dra igång med collector-modulen redan nu. Vi pratade igenom allt och kom fram till att jag och Malin kommer stå för att traversera “levande” stackpekare som ska skickas till de som har hand om heappekarna.
Tid nedlagd: 1h

Implementering
Vi satt med detta på kvällen och lekte runt med att printa olika saker som stackaddresser osv för att få en förståelse för hur vi ska och kan göra detta.
Tid nedlagd: 6h

16/12 Implementering
Vi letade ett bra sätt att hitta stackens topaddress idag. Gjorde inga direkt framsteg men vi började skriva på en funktion som skapar en länkad lista som vi planerar att lägga de “levande” stackpekarna i som sedan ska skickas vidare i programmet. Just nu skapar den bara en lista och vi sätter in alla stackpekare i den och printar ut det för att hitta eventuella rätt/fel/nya saker vi inte visste.
Tid nedlagd: 6h

18/12 Implementering/testning
Jag lekte runt lite till med koden för att förstå den bättre. Den börjar printa ut mer rätta saker än fel så framsteg görs. Har fortfarande inte kommit fram till hur vi ska kolla om en pekare är “levande” eller inte.
Tid nedlagd: 2h

20/12 Testning
Jag satte mig med koden och kom på lite bättre sätt att printa saker och ting. Fortfarande inga märkbara framsteg dock…
Tid nedlagd: 3h

21/12 Implementering
Jag och Malin satte oss ner på Studentpalatset i Stockholm och har hittills gjort fina framsteg. Vi pratade med Carl som sa att han tänker göra en funktion som kollar heapobjektens metadata för att avgöra om pekaren pekar på ett objekt eller inte, så vi behöver inte tänka på den biten. Vi har utökat koden lite så att den bara ska lägga till de stackpekare som klarar metadatakollen (***boolfunktion*** *så vi gjorde en* ***if-sats*** *för den*). Vi bör fortfarande lära oss mer om hur man finner att en stackpekare pekar på ett objekt som fortfarande behövs i programmet och det kommer vi tackla den kommande tiden.
Tid nedlagd: 6h

23/12 Implementering, testning
Vi har fixat några funktioner för att kolla om en stackpekare pekar på vår heap, hur heapen växer och hur stacken växer. Vi försökte få till testerna på de tre olika plattformarna att fungera korrekt. Vi har kodat så att den ska printa ut varje stackpekare som läggs till i vår länkade lista och detta fungerar när vi kompilerar filen på våra egna datorer, men när vi körde testerna på de olika plattformarna så printade den inte ut all info, och utelämnade det väsentliga. Efter en del korrigeringar kördes det korrekt på två av tre plattformar, vilket är positivt.
Tid nedlagd: 5h

25/12 Implementering, design, testning
Jag satte mig ner själv och ändrade lite funktionsnamn och skrev några nya rader i main-funktionen som ska testa modulen på ett bättre sätt. Dessa rader kräver dock lite mer från heap-modulen än det som finns i dagsläget så dessa tester blir relevanta snart.
Tid nedlagd: 30min

27/12 Testning
Vi har läst på om enhetstester idag för att skapa en testmodul till vår stacktraverserare. Det var lite svårare än förväntat då ingen av oss faktiskt har kommit i kontakt med att skriva enhetstester i tidigare arbeten. Fram tills denna logg skrevs har vi läst på om det hela på CUnits hemsida och i *Learn C the hard way*. Vi kommer fortsätta läsa på om detta idag och imorgon och försöka behärska detta tills övermorgon den **29 december**.
Tid nedlagd: 4h

28/12 Teori
Idag läste jag på om enhetstester och försökte förstå mig på CUnit bättre. Jag gjorde några övningar i *Learn C The Hard Way* vilket tog lite mer tid än förväntat men jag kände att jag fick lite bättre förståelse för det hela.
Tid nedlagd: 5h

29/12 Testning
Vi skrev klart de flesta testerna idag. Själva kompileringen krånglade så Malin tog sig an att kolla upp och försöka lösa det medan jag ska fortsätta skriva tester.
Tid nedlagd: 5h

31/12 Implementation
Jag skrev till lite fler tester idag. Malin hade fortfarande inte hittat en lösning till makefilen så jag hjälpte till men hittade inte heller en bra lösning tyvärr.. Har varit frustrerande att koda idag minst sagt. Nu ska jag fira nyår!
Tid nedlagd: 4h

2/1 Testning
Jag frågade Carl om hjälp med makefilen och han lyckades lösa det åt oss. Dessvärre råkade jag spilla kaffe på min laptop stunden senare vilket har förstört mitt tangentbord på min laptop... Hittade till slut ett externt tangentbord som några släktingar gav mig men det kom en del personliga saker i vägen för kodandet på kvällen.
Tid nedlagd: 2h

4/1 Implementation
Idag upptäckte jag lite fel i vår kod i originalfilen då testfilen inte klarade alla assertions. Upptäckte att vi hade glömt att ***int-casta*** vissa ***voidpekare*** och lite konstigheter då funktionen som kollar om en stackpekares innehåll pekar i vår heap inte la till alla pekare som uppfyllde kravet. Känns som att detta kan bero på att vi inte tagit ***data-alignment*** i åtanke än.. Jag och Malin ska titta på det tillsammans imorgon.
Tid nedlagd: 5h

5/1 Implementation
Forskade runt kring data alignment och försökte få en känsla för det och se om det har påverkat vårt arbete.

- Det verkar som att C har inbyggt att anpassa koden efter datorns alignment.
- Hade även problem med vår heaps adresser då dess **end-address** just nu är 0 när den **int-castas**. Detta leder till att alla tal mellan 0 och det skyhöga talet som heapens **start-address** är kommer betraktas som pekare som pekar i vår heap. Vi hade pratat om att ha höga addresser för heapen just för att undvika detta problem. Väntar på svar från heapgänget angående det.
- Stötte även på en del kompileringsfel på de olika plattformarna vilket är något vi ska titta på imorgon.
- Ska slutligen titta på funktionerna vi har i vår modul på tåget på vägen hem och ge dem bättre namn om det behövs.

Det blev en del att göra och tänka på idag.
Tid nedlagd: 7h

6/1 Implentation
Jag tänkte kolla på alignmentskillnader mellan de olika plattformarna medan Malin tog på sig att försöka lösa våra kompileringsproblem. Jag fann tyvärr att min kladdfil inte kan köras på **SPARC** och **Solaris**… Har googlat ett bra tag nu men inte hittat en bra lösning på detta problem. Ska fråga på Piazza. Var tvungen att jobba med en annan uppgift tidigare på dagen så jag hann tyvärr inte med så mycket som jag hade önskat.
Tid nedlagd: 4h

7/1 Implementation
Kompileringsproblemen på de olika plattformarna berodde på att jag hade kompilerat den exekverbara filen på min plattform och sedan försökt köra den på andra plattformar. Lösningen visade sig vara så enkel att jag bara behövde radera den ***<min plattform>***-kompilerade filen och kompilera om den på respektive plattform så gick det att köra! Slösade lite väl mycket tid på detta känner jag… Efter att detta löst sig fortsatte vi att kämpa med type castingen på void pekarna för att jämföra stackaddressernas innehåll med heapens start och end addresser för att hitta pekare som pekar i heapen. Det tog väldigt lång tid och vi vet inte riktigt om det är problem med type castingen eller om pekaren vi allokerat i vår heap inte allokeras. Har ett personligt ärende imorgon som kommer hålla mig borta från plugget och Uppsala tills kvällen tyvärr, Blir en del att bita i i helgen…
Tid nedlagd: 8h

9/1 Implementation
Kompileringsproblemen på den egna plattformen löste Malin, och jag städade upp **stack_traverser.c** så den inte innehåller onödig kod. Det visade sig att problemet vi hade med att hitta heappekarna i stacken inte berodde på typecastingen, utan det berodde på alignmenten trots allt. Vi löste det med ett tips och förstod att vi hade glömt att man kan kolla varje **byte** på stacken genom att inkrementera pekaren till toppen med en byte istället för **sizeof(void *)** som vi hade innan (vilken inkrementerade pekaren med 8 bytes). Äntligen var detta problem löst, som det hemsökte oss… Detta leder förstås till att programmet blir 8 gånger långsammare, men det kan vi leva med i nuläget tills vi hittar en effektivare lösning. Imorgon ska vi samlas med

gruppen och sitta tillsammans hela dagen, vilket är skönt då vi inte haft möjlighet till att göra det sen starten av projektet.
Tid nedlagd: 5h

10/1 Implementation, Testning
Vi försökte oss på att införa lite dubbelpekare här och där och fann till slut att vi inte la till heapaddresserna i den länkade listan utan vi la till stackaddresserna som pekade på heapaddresserna. Så dubbelpekare blev händigt här då skapade en dubbelpekare som tar emot stackaddressen och sen avrefererar vi den när vi skickar in den i den länkade listans insättningsfunktion så blir allting rätt. Vi ändrade även print-funktionen till bara några rader, så allt ser snyggt ut nu. Återstår att få testfilen att fungera och att framställa själva skräpsamlaren.
Tid nedlagd: 9h

11/1 Implementation, Design, Testning, Postmortem
Alla utom en gruppmedlem (som befinner sig i Singapore), samlades idag och knöt samman allas koder. Inte helt oväntat uppkom det massvis med errors men detta åtgärdades som tur efter några timmar. Jag och Malin var tvungna att springa iväg och skriva kodprov kl. 13-16, och under den tiden försvann alla errors. När vi väl kom tillbaka upptäckte vi att stacktraverseraren hamande i en oändlig loop när den kördes tillsammans med alla andras moduler, konstigt nog. Efter nästan en timmes GDB:ande från min sida hittade Victor felet; variabeln ***environ*** som ska vara stackens bottenaddress var inte deklarararerad som ***extern*** i en av de andra modulerna, vilket mixtrade med variabelvärdet på ***extern char/voidv **environ*** i varje modul som inkluderade denna modul där den inte var externt deklarerad.
Efter det fixade jag de återstående testerna för vår modul och började sedan på den individuella reflektionen.
Slutligen började jag på prestandatester och ska göra klart dem imorgon.
Tid nedlagd: 13h

12/1 Implementation, Design, Testning, Dokumentation, Postmortem
Idag fortsatte jag kort på prestandatesterna men fick ta över uppgiften att implementera skräpsamlaren på ett redan fungerande program för att testa. Dock fick jag kort därpå ta mig an uppgiften att skriva vår rapport och se till att den blir klar så snabbt som möjligt. Gjorde det mesta och lämnade över varje del som tillhör en person till den personen att skriva klart. Skräpsamlaren fungerade till 90% (den kompakterar och byter adresser på heapobjekten och pekar om stackpekarna till den nya addressen men objekten som det förflyttade heapobjektet pekar på flyttas ännu inte över till rätt address, och man måste allokera saker och ting i exakt den ordningen man har tänk att de ska ligga i minnet. Annars knasar samlaren ur på den punkten. Förövrigt fungerar den som den ska!!)
Tid nedlagd: 10h

Total tid nedlagd: 122h