; Nicholas Martinez

; COSCI 416 - M 5:45 - 10:00PM

; M. Rettke

; May 28th 2018

; Final Exam

;

;File: fig0648.pep

;Computer Systems, Fifth edition

;Figure 6.48

;

BR main

data: .EQUATE 0 ;struct field #2d

next: .EQUATE 2 ;struct field #2h

p2check: .BLOCK 2

;

;\*\*\*\*\*\*\* main ()

p2: .EQUATE 8 ; local #2h

first2: .EQUATE 6 ; local #2h

first: .EQUATE 4 ;local variable #2h

p: .EQUATE 2 ;local variable #2h

value: .EQUATE 0 ;local variable #2d

rmsg: .ASCII "In reverse:\n\x00"

;; Original Program

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

main: SUBSP 10,i ;push #p2 #first2 #first #p #value

LDWA 0,i ;first = 0

STWA first,s

DECI value,s ;scanf("%d", &value);

while: LDWA value,s ;while (value != -9999)

CPWA -9999,i

BREQ endWh

LDWA first,s ;p = first

STWA p,s

LDWA 4,i ;first = (struct node \*) malloc(sizeof(struct node))

CALL malloc ;allocate #data #next

STWX first,s

LDWA value,s ;first->data = value

LDWX data,i

STWA first,sfx

LDWA p,s ;first->next = p

LDWX next,i

STWA first,sfx

DECI value,s ;scanf("%d", &value)

BR while

endWh: LDWA first,s ;for (p = first

STWA p,s

for: LDWA p,s ;p != 0

CPWA 0,i

BREQ endFor

LDWX data,i ;printf("%d ", p->data)

DECO p,sfx

LDBA ' ',i

STBA charOut,d

LDWX next,i ;p = p->next)

LDWA p,sfx

STWA p,s

BR for

endFor: BR reverse

STOP

;; Original Program End

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Addition for Final

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

reverse: LDWA 0,i ; Begin reverse sequence

STWA first2,s ; Set first2 = 0

LDWA first,s ; p = first

STWA p,s

for2: LDWA p,s ; Check address of p for null (next member)

ADDA 2,i ; Adding 2 to heap pointer location for next member

STWA p2check,d ; Store this value in global as to not change the

LDWA p2check,n ; actual p pointer, load the next value into accumulator

CPWA 0,i ; Determine if were at end of list, p->next == 0

BREQ lastrun ; If so, jump to last iteration, since we need to get the last data

LDWA first2,s ; p2 = first2

STWA p2,s ;

LDWA 4,i ; first2 = malloc()

CALL malloc ; allocate #data #next

STWX first2,s

LDWA p,sf ;first->next = p

STWA first2,sf ;Store value of first list to second list

LDWA p2,s ; Set first2->next to p2

LDWX first2,s ; Access first2 next member by increment it by 2

ADDX 2,i ;

STWX first2,s ; Store address back to first2

LDWA p2,s ; Load value of p2 to accumulator

STWA first2,sf ; Store at first2->next

LDWA p,s ; Traverse the heap in reverse

SUBA 4,i ; By moving directly to next data item

STWA p,s

BR for2

lastrun: LDWA first2,s ; p2 = first2

STWA p2,s

LDWA 4,i ; first2 = malloc()

CALL malloc ; allocate #data #next

STWX first2,s

LDWA p,sf ;first->next = p

STWA first2,sf ;Store value of first list to second list

LDWA p2,s ; Set first2->next to p2

LDWX first2,s ; Access first2 next member by increment it by 2

ADDX 2,i ;

STWX first2,s ; Store address back to first2

LDWA p2,s ; Load value of p2 to accumulator

STWA first2,sf ; Store at first2->next

LDWA p,s ; Traverse the heap in reverse

SUBA 4,i ; By moving directly to next data item

STWA p,s

LDBA '\n',i

STBA charOut,d

LDBA '\n',i

STBA charOut,d

STRO rmsg,d

prntrvrs:LDWA first2,s ; Load first2 which should be pointing to last next member

SUBA 2,i ; Subtract 2 to get to data member

STWA first2,s ; Store data member address back to stack

DECO first2,sf ; Output the contents of data member address

LDBA ' ',i ; Print space

STBA charOut,d

LDWA first2,s ; Reload first2 (data member address)

ADDA 2,i ; Add 2 to get back to next member

STWA first2,s ; Store next member address of stack

LDWX first2,sf ; Load the contents ofaddress of the next member into index

STWX first2,s ; Store that back on to stack

LDWA first2,s ; Check to see if its the last member (0000 next address)

CPWA 0,i ; If it is, end the program

BREQ endfor2 ;

BR prntrvrs ; Otherwise start loop again

endfor2: STOP

;; End Addition for Final

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;

;\*\*\*\*\*\*\* malloc()

; Precondition: A contains number of bytes

; Postcondition: X contains pointer to bytes

malloc: LDWX hpPtr,d ;returned pointer

ADDA hpPtr,d ;allocate from heap

STWA hpPtr,d ;update hpPtr

RET

hpPtr: .ADDRSS heap ;address of next free byte

heap: .BLOCK 1 ;first byte in the heap

.END

ASSEMBLER LISTING

-------------------------------------------------------------------------------

Object

Addr code Symbol Mnemon Operand Comment

-------------------------------------------------------------------------------

;File: fig0648.pep

;Computer Systems, Fifth edition

;Figure 6.48

;

0000 120012 BR main

data: .EQUATE 0 ;struct field #2d

next: .EQUATE 2 ;struct field #2h

0003 0000 p2check: .BLOCK 2

;

;\*\*\*\*\*\*\* main ()

p2: .EQUATE 8 ; local #2h

first2: .EQUATE 6 ; local #2h

first: .EQUATE 4 ;local variable #2h

p: .EQUATE 2 ;local variable #2h

value: .EQUATE 0 ;local variable #2d

0005 496E20 rmsg: .ASCII "In reverse:\n\x00"

726576

657273

653A0A

00

0012 58000A main: SUBSP 10,i ;push #p2 #first2 #first #p #value

0015 C00000 LDWA 0,i ;first = 0

0018 E30004 STWA first,s

001B 330000 DECI value,s ;scanf("%d", &value);

001E C30000 while: LDWA value,s ;while (value != -9999)

0021 A0D8F1 CPWA -9999,i

0024 18004E BREQ endWh

0027 C30004 LDWA first,s ;p = first

002A E30002 STWA p,s

002D C00004 LDWA 4,i ;first = (struct node \*) malloc(sizeof(struct node))

0030 240137 CALL malloc ;allocate #data #next

0033 EB0004 STWX first,s

0036 C30000 LDWA value,s ;first->data = value

0039 C80000 LDWX data,i

003C E70004 STWA first,sfx

003F C30002 LDWA p,s ;first->next = p

0042 C80002 LDWX next,i

0045 E70004 STWA first,sfx

0048 330000 DECI value,s ;scanf("%d", &value)

004B 12001E BR while

004E C30004 endWh: LDWA first,s ;for (p = first

0051 E30002 STWA p,s

0054 C30002 for: LDWA p,s ;p != 0

0057 A00000 CPWA 0,i

005A 180075 BREQ endFor

005D C80000 LDWX data,i ;printf("%d ", p->data)

0060 3F0002 DECO p,sfx

0063 D00020 LDBA ' ',i

0066 F1FC16 STBA charOut,d

0069 C80002 LDWX next,i ;p = p->next)

006C C70002 LDWA p,sfx

006F E30002 STWA p,s

0072 120054 BR for

0075 120079 endFor: BR reverse

0078 00 STOP

0079 C00000 reverse: LDWA 0,i ; Begin reverse sequence

007C E30006 STWA first2,s ; Set first2 = 0

007F C30004 LDWA first,s ; p = first

0082 E30002 STWA p,s

0085 C30002 for2: LDWA p,s ; Check address of p for null (next member)

0088 600002 ADDA 2,i ; Adding 2 to heap pointer location for next member

008B E10003 STWA p2check,d ; Store this value in global as to not change the

008E C20003 LDWA p2check,n ; actual p pointer, load the next value into accumulator

0091 A00000 CPWA 0,i ; Determine if were at end of list, p->next == 0

0094 1800CA BREQ lastrun ; If so, jump to last iteration, since we need to get the last data

0097 C30006 LDWA first2,s ; p2 = first2

009A E30008 STWA p2,s ;

009D C00004 LDWA 4,i ; first2 = malloc()

00A0 240137 CALL malloc ; allocate #data #next

00A3 EB0006 STWX first2,s

00A6 C40002 LDWA p,sf ;first->next = p

00A9 E40006 STWA first2,sf ;Store value of first list to second list

00AC C30008 LDWA p2,s ; Set first2->next to p2

00AF CB0006 LDWX first2,s ; Access first2 next member by increment it by 2

00B2 680002 ADDX 2,i ;

00B5 EB0006 STWX first2,s ; Store address back to first2

00B8 C30008 LDWA p2,s ; Load value of p2 to accumulator

00BB E40006 STWA first2,sf ; Store at first2->next

00BE C30002 LDWA p,s ; Traverse the heap in reverse

00C1 700004 SUBA 4,i ; By moving directly to next data item

00C4 E30002 STWA p,s

00C7 120085 BR for2

00CA C30006 lastrun: LDWA first2,s ; p2 = first2

00CD E30008 STWA p2,s

00D0 C00004 LDWA 4,i ; first2 = malloc()

00D3 240137 CALL malloc ; allocate #data #next

00D6 EB0006 STWX first2,s

00D9 C40002 LDWA p,sf ;first->next = p

00DC E40006 STWA first2,sf ;Store value of first list to second list

00DF C30008 LDWA p2,s ; Set first2->next to p2

00E2 CB0006 LDWX first2,s ; Access first2 next member by increment it by 2

00E5 680002 ADDX 2,i ;

00E8 EB0006 STWX first2,s ; Store address back to first2

00EB C30008 LDWA p2,s ; Load value of p2 to accumulator

00EE E40006 STWA first2,sf ; Store at first2->next

00F1 C30002 LDWA p,s ; Traverse the heap in reverse

00F4 700004 SUBA 4,i ; By moving directly to next data item

00F7 E30002 STWA p,s

00FA D0000A LDBA '\n',i

00FD F1FC16 STBA charOut,d

0100 D0000A LDBA '\n',i

0103 F1FC16 STBA charOut,d

0106 490005 STRO rmsg,d

0109 C30006 prntrvrs:LDWA first2,s ; Load first2 which should be pointing to last next member

010C 700002 SUBA 2,i ; Subtract 2 to get to data member

010F E30006 STWA first2,s ; Store data member address back to stack

0112 3C0006 DECO first2,sf ; Output the contents of data member address

0115 D00020 LDBA ' ',i ; Print space

0118 F1FC16 STBA charOut,d

011B C30006 LDWA first2,s ; Reload first2 (data member address)

011E 600002 ADDA 2,i ; Add 2 to get back to next member

0121 E30006 STWA first2,s ; Store next member address of stack

0124 CC0006 LDWX first2,sf ; Load the contents ofaddress of the next member into index

0127 EB0006 STWX first2,s ; Store that back on to stack

012A C30006 LDWA first2,s ; Check to see if its the last member (0000 next address)

012D A00000 CPWA 0,i ; If it is, end the program

0130 180136 BREQ endfor2 ;

0133 120109 BR prntrvrs ; Otherwise start loop again

0136 00 endfor2: STOP

;

;\*\*\*\*\*\*\* malloc()

; Precondition: A contains number of bytes

; Postcondition: X contains pointer to bytes

0137 C90141 malloc: LDWX hpPtr,d ;returned pointer

013A 610141 ADDA hpPtr,d ;allocate from heap

013D E10141 STWA hpPtr,d ;update hpPtr

0140 01 RET

0141 0143 hpPtr: .ADDRSS heap ;address of next free byte

0143 00 heap: .BLOCK 1 ;first byte in the heap

0144 .END

-------------------------------------------------------------------------------

Symbol table

--------------------------------------

Symbol Value Symbol Value

--------------------------------------

data 0000 endFor 0075

endWh 004E endfor2 0136

first 0004 first2 0006

for 0054 for2 0085

heap 0143 hpPtr 0141

lastrun 00CA main 0012

malloc 0137 next 0002

p 0002 p2 0008

p2check 0003 prntrvrs 0109

reverse 0079 rmsg 0005

value 0000 while 001E

--------------------------------------

Object Code

12 00 12 00 00 49 6E 20 72 65 76 65 72 73 65 3A

0A 00 58 00 0A C0 00 00 E3 00 04 33 00 00 C3 00

00 A0 D8 F1 18 00 4E C3 00 04 E3 00 02 C0 00 04

24 01 37 EB 00 04 C3 00 00 C8 00 00 E7 00 04 C3

00 02 C8 00 02 E7 00 04 33 00 00 12 00 1E C3 00

04 E3 00 02 C3 00 02 A0 00 00 18 00 75 C8 00 00

3F 00 02 D0 00 20 F1 FC 16 C8 00 02 C7 00 02 E3

00 02 12 00 54 12 00 79 00 C0 00 00 E3 00 06 C3

00 04 E3 00 02 C3 00 02 60 00 02 E1 00 03 C2 00

03 A0 00 00 18 00 CA C3 00 06 E3 00 08 C0 00 04

24 01 37 EB 00 06 C4 00 02 E4 00 06 C3 00 08 CB

00 06 68 00 02 EB 00 06 C3 00 08 E4 00 06 C3 00

02 70 00 04 E3 00 02 12 00 85 C3 00 06 E3 00 08

C0 00 04 24 01 37 EB 00 06 C4 00 02 E4 00 06 C3

00 08 CB 00 06 68 00 02 EB 00 06 C3 00 08 E4 00

06 C3 00 02 70 00 04 E3 00 02 D0 00 0A F1 FC 16

D0 00 0A F1 FC 16 49 00 05 C3 00 06 70 00 02 E3

00 06 3C 00 06 D0 00 20 F1 FC 16 C3 00 06 60 00

02 E3 00 06 CC 00 06 EB 00 06 C3 00 06 A0 00 00

18 01 36 12 01 09 00 C9 01 41 61 01 41 E1 01 41

01 01 43 00 zz