**CSE 474 – Phase 1a**

CFG

< program >      🡪 #Start **OnceUponATime** < statement list > **TheEnd**

< statement list > 🡪 < statement > *{ < statement > }*

< statement >      🡪 < ident > **:=** < expression > **;** #Assign

< statement >      🡪 < ident > **:=** < string > **;** #Assign

< statement > 🡪 < ident > **;** #DeclareNotAssign

< statement >     🡪 **READ (** < id list > **) ;**

< statement >      🡪 **WRITE (** < expr list > **) ;**

< statement >      🡪 **WRITE (** < string > **) ;**

< id list >      🡪 < ident > *{****,*** *< ident > }* #ReadId

< expr list >      🡪 < expression > *{****,*** *< expression > }* #WriteExpr

< string >     🡪 **“** < char list > **”** *{* ***+******“*** *< char list >* ***”*** *#Concat }*

< expression > 🡪 < primary > *{ < add op > < primary > #GenInfix }*

< primary >      🡪 **(** < expression > **)**

< primary >     🡪 < ident >

< primary >     🡪 IntLiteral #ProcessLiteral

< add op >      🡪 PlusOp #ProcessOp

< add op >      🡪 MinusOp #ProcessOp

< char list >      🡪 *{* ***asci character*** *}* #ZeroOrMoreCharacters

< ident >          🡪 < data type > **Id** #ProcessId

< ident >          🡪 **Id** #ProcessId #MustBeInSymbolTable

< data type > 🡪 **~i** #Int

< data type > 🡪 ~**s** #String

< system goal >     🡪 < program > EofSym #Finish

Test Cases

Integer declaration

**OnceUponATime**

**~i x;**

**TheEnd**

.text

.global \_start

\_start:

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

x: .int 0

Integer declaration with runtime assignment

**OnceUponATime**

**~i x;**

**x := 5;**

**TheEnd**

.text

.global \_start

\_start:

mov $5, %eax;

mov %eax, x;

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

x: .int 0

Integer declaration / initialization

**OnceUponATime**

**~i x := 5;**

**TheEnd**

.text

.global \_start

\_start:

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

x: .int 5

Integer declaration / initialization via addition

**OnceUponATime**

**~i x := 5 + 3;**

**TheEnd**

.text

.global \_start

\_start:

addl $3, %eax

movl %eax, x

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

x: .int 5

String declaration and runtime assignment

**OnceUponATime**

**~s a;**

**a := "Hello";**

**TheEnd**

.text

.global \_start

\_start:

\*loop to move \_temp into a\*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

a: .zero 256

\_temp: .string “Hello”

String declaration / initialization

**OnceUponATime**

**~s a := "Hello";**

**TheEnd**

.text

.global \_start

\_start:

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

a: .string "Hello"

String concatenation on declaration

**OnceUponATime**

**~s b := "Hel" + "lo";**

**TheEnd**

.text

.global \_start

start:

\* loop to move b into \_temp2 \*

\* loop to move \_temp1 into temp2 \*

\* assign \_temp2 to b \*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

b: .string "Hel"

\_temp1: .string “lo”

\_temp2: .zero 256

String concatenation via 1 variable

**OnceUponATime**

**~s a := "llo";**

**~s c := "He" + a;**

**TheEnd**

.text

.global \_start

\_start:

\*move c into \_temp\*

\*move a into \_temp\*

\*move \_temp into c\*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

a: .string "llo"

c: .string "He"

\_temp .zero 256

String concatenation via 2 variables

**OnceUponATime**

**~s a := "He”;**

**~s c := "llo";**

**~s b := a + b;**

**TheEnd**

.text

.global \_start

\_start:

\*move a into b\*

\*move c into b\*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

a: .string "He"

c: .string "llo"

b: .zero 256

Read and write integers

**OnceUponATime**

**~t y;**

**READ ( y );**

**~t z := y + 1;**

**WRITE( z );**

**TheEnd**

.text

.global \_start

\_start:

\* read ( y ) \*

mov y, %eax

mov %eax, z

addl $1, z

\* write ( z ) \*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

y: .int 0

z: .int 0

Read strings

**OnceUponATime**

**~s y;**

**READ ( y );**

**TheEnd**

.text

.global \_start

\_start:

\* read ( y ) \*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

y: .zero 256

Write strings

**OnceUponATime**

**~s y = “Hello”;**

**WRITE ( y );**

**TheEnd**

.text

.global \_start

\_start:

\* write ( y ) \*

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

y: .string “Hello”

ERROR – Variable assignment without declaration

**OnceUponATime**

**c := 5**

**TheEnd**

.text

.global \_start

\_start:

ERROR - ‘c’ not declared

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

ERROR – String concatenation without variable declaration

**OnceUponATime**

**~s c := "He" + a;**

**TheEnd**

.text

.global \_start

\_start:

ERROR – ‘a’ not declared

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data

c: .string "He"

ERROR – Read input to undeclared variable

**OnceUponATime**

**READ ( y );**

**TheEnd**

.text

.global \_start

\_start:

ERROR – ‘y’ not declared

exit:

mov $1, %eax

mov $1, %ebx

int $0x80

.data