JSica: 타이젠앱 개인정보 누출 분석

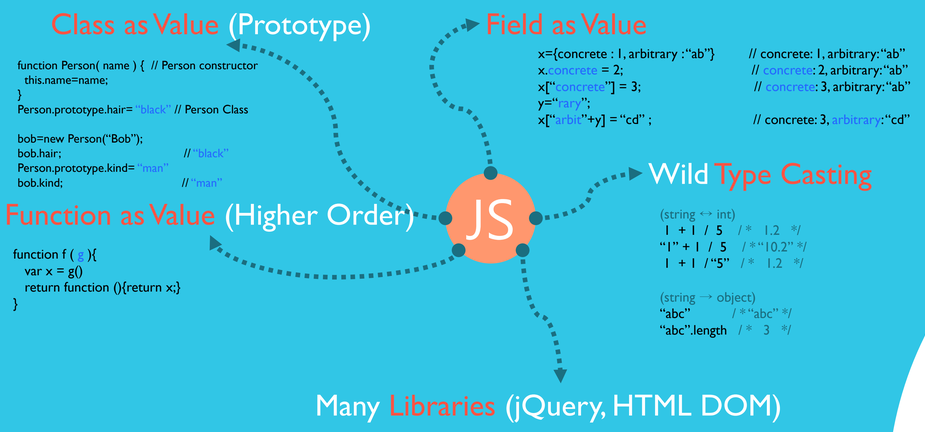
강동옥 강지훈 이광근
서울대학교프로그래밍연구실

Programming Research Laboratory
SEOUL NATIONAL UNIVERSITY

타이젠 앱과 자바스크립트

- 타이젠 앱 구성요소

- 자유분방한 자바스크립트



결과

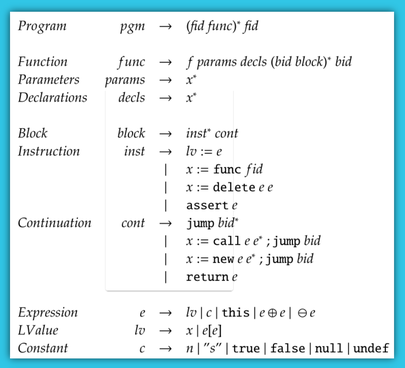
타이젠 앱	Code Size(LoC)	Time(s)	Mem (MB)	Alarm(#)	Leak Info.
analogWatch	96	1	23		
audioFilter	329	3	23		
bluetoothChat	1404	1	34	3	Bluetooth
callLog	1481	1	34		
chatter	1578	1	35	6	Phone Number
compass	84	1	23		
contactsExchanger	1394	1	34	42	Phone Number, NFC
deviceMotionCapture	86	1	23		
eventManager	1664	1	57		
exercisePlanner	1787	3	39		
fileManager	2030	1	35		
imageRotation	34	2	23		
mediaContent	262	1	34		
npRuntime	115	1	23		
offlineClockImage	105	2	23		
piano	130	1	23		
selfcamera	686	1	34	3	Photo
sensorBall	532	1	23		
systemInfo	179	1	34		
taskManager	224	6	23		
touchPaint	117	9	34		

Sunspider1.0.3 벤치마크	Code Size (LoC)	Time (s)	Mem (MB)
3d-cube	355	2	35
3d-morph	63	1	35
3d-raytrace	447	1	35
access-binary-trees	54	1	35
access-nbody	174	1	35
access-nsieve	46	1	20
bitops-3bit-bits-in-byte	40	1	20
bitops-bits-in-byte	30	1	20
bitops-bitwise-and	35	1	20
bitops-nsieve-bits	42	1	20
controlflow-recursive	32	1	20
crypto-md5	292	1	35
crypto-sha1	228	1	35
math-cordic	106	1	35
math-partial-sums	45	1	35
math-spectral-norm	60	1	35
regexp-dna	1721	1	42
string-base64	137	1	35
string-fasta	90	4	39

- 대부분의 타이젠 샘플앱과 Sunspider 벤치마크가 **1초내외** 분석
- 샘플앱에 대해서는 허위경보 없음

디자인

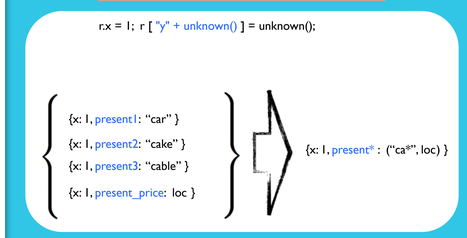
- JSica Core



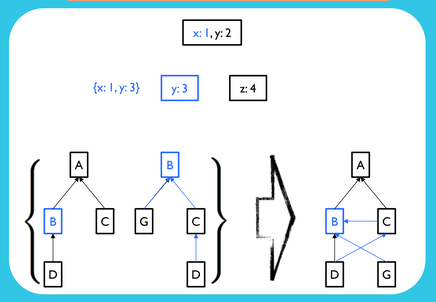
$$
\begin{array}{lrcl}
\textit{Program} & pgm & \to & (\textit{fid}\ \textit{func})^*\ \textit{fid} \\
\\
\textit{Function} & \textit{func} & \to & f\ \textit{params}\ \textit{decls}\ (\textit{bid}\ \textit{block})^*\ \textit{bid} \\
\textit{Parameters} & \textit{params} & \to & x^* \\
\textit{Declarations} & \textit{decls} & \to & x^* \\
\\
\textit{Block} & \textit{block} & \to & \textit{inst}^*\ \textit{cont} \\
\textit{Instruction} & \textit{inst} & \to & \textit{lv} := e \\
 & & | & x := \texttt{func}\ \textit{fid} \\
 & & | & x := \texttt{delete}\ e\ e \\
 & & | & \texttt{assert}\ e \\
\textit{Continuation} & \textit{cont} & \to & \texttt{jump}\ \textit{bid}^* \\
 & & | & x := \texttt{call}\ e\ e^*\ ;\ \texttt{jump}\ \textit{bid} \\
 & & | & x := \texttt{new}\ e\ e^*\ ;\ \texttt{jump}\ \textit{bid} \\
 & & | & \texttt{return}\ e \\
\\
\textit{Expression} & e & \to & \textit{lv} \mid c \mid \texttt{this} \mid e \oplus e \mid \ominus e \\
\textit{LValue} & \textit{lv} & \to & x \mid e[e] \\
\textit{Constant} & c & \to & n \mid \text{"}s\text{"} \mid \texttt{true} \mid \texttt{false} \mid \texttt{null} \mid \texttt{undef}
\end{array}
$$

$$
\begin{array}{rcl}
\hat{T} \in \hat{Trace} & = & \Delta \xrightarrow{fin} \hat{State} \\
\hat{S} \in \hat{State} & = & \hat{Heap} \times \hat{Store} \times \hat{This} \\
\hat{H} \in \hat{Heap} & = & (\hat{Loc} \xrightarrow{fin} \hat{Obj}) \times (\hat{Loc} \xrightarrow{fin} 2^{\hat{Loc}}) \\
\hat{M} \in \hat{Store} & = & \hat{Var} \xrightarrow{fin} \hat{Val} \\
\hat{th} \in \hat{This} & = & 2^{\hat{Loc}} \\
\\
\hat{o} \in \hat{Obj} & = & \hat{Closure} \times \hat{Record} \\
\hat{\rho} \in \hat{Record} & = & \hat{Str} \xrightarrow{fin} (\hat{Val} \times \hat{Bool}) \\
\hat{cls} \in \hat{Closure} & = & \hat{Fid} \xrightarrow{fin} \hat{Bindings} \\
\\
\hat{v} \in \hat{Val} & = & 2^{\hat{Loc}} \times \hat{Bool} \times \hat{Num} \times \hat{Str} \times \{\text{undef}, \text{null}\}_\perp \\
\hat{u} \in \hat{Loc} & = & AllocSite \\
\hat{s} \in \hat{Str} & = & Str + Prefix
\end{array}
$$

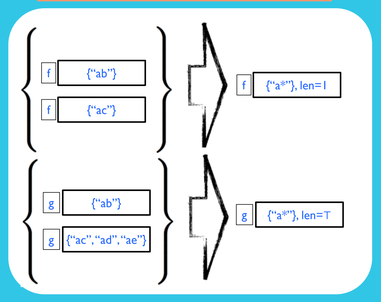
Object Record



Prototype Chain



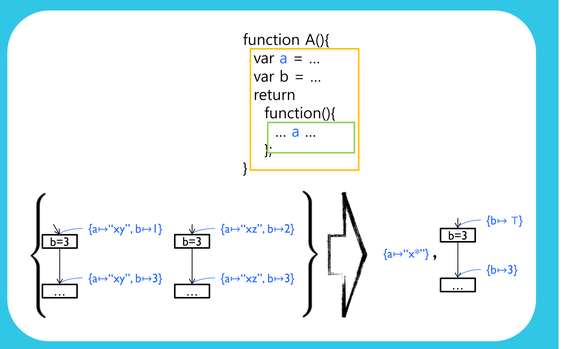
Partial Binding



최적화

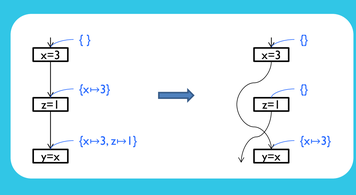
$$\hat{T} \in \hat{Trace} \quad = \quad \hat{Heap} \times \hat{WStore} \times (\Delta \xrightarrow{fin} \hat{Store} \times \hat{This})$$

- 정확도 개선: **Strong 메모리와 Weak 메모리를 구분하는 요약**



- 속도 개선:
  Weak 업데이트 메모리는 **경로 둔감 분석**(Flow Insensitive)
  Strong 업데이트 메모리는 **스파스 분석**(Sparse Analysis)

Def-Use 그래프를 업데이트 하며 필요한 지점에 필요한 메모리만 넘겨줌



포스터 디자인 : 9th 윌샵 이원찬 연구원님의 포스터를 참조