

Protocolo de comunicación - Draft

Milton Inostroza Aguilera

28 Mayo de 2008

Resumen

Protocolo de comunicación utilizado para enviar datos desde pyTOD hacia TOD.

Indice

1	Protocolo de comunicación	4
1.1	Identificadores	4
1.1.1	Sucesos	4
1.1.2	Objetos	4
1.1.3	Tipo de datos	5
1.2	Registro de objetos	5
1.2.1	Función	5
1.2.2	Variable local	5
1.2.3	Clase	6
1.2.4	Método	6
1.2.5	Atributo	6
1.2.6	Thread	6
1.2.7	Probe	7
1.3	Llamada de objetos	7
1.3.1	Función	7
1.3.2	Método	7
1.4	Instanciación de clase	8
1.5	Asignación - Modificación de objetos	8
1.5.1	Variable local	8
1.5.2	Atributo	8
1.6	Return	9

Lista de Tablas

1	Identificadores de sucesos	4
2	Identificadores de objetos	4
3	Identificadores de tipo de datos	5
4	Registro del objeto función	5
5	Registro del objeto variable local	5
6	Registro del objeto clase	6
7	Registro del objeto método	6
8	Registro del objeto atributo	6
9	Registro del objeto thread	6
10	Registro del objeto probe	7
11	Llamada al objeto función	7
12	Llamada al objeto método	7
13	Coordenadas	7
14	Instanciación de clase	8
15	Registro del objeto variable local	8
16	Registro del objeto atributo	8
17	Registro de return	9

1 Protocolo de comunicación

Se muestran los identificadores bases para el registro de los objetos, envío de eventos y tipo de datos:

1.1 Identificadores

1.1.1 Sucesos

La siguiente tabla muestra que cada suceso tiene un identificador en el sistema de captación de huella.

Suceso	Identificador
Registro	0
Llamada	1
Asignación	2
Retorno	3
Instanciación	4

Tabla 1: Identificadores de sucesos

1.1.2 Objetos

La siguiente tabla muestra que cada objeto tiene un identificador en el sistema de captura de huella.

Id Objeto	Identificador
Clase	0
Método	1
Atributo	2
Función	3
Variable local	4
Probe	5
Thread	6

Tabla 2: Identificadores de objetos

1.1.3 Tipo de datos

La siguiente tabla muestra que cada tipo de datos tiene un identificador en el sistema de captura de huella.

Type	Identificador
int	0
str	1
float	2
long	3
bool	4
list	5
tuple	6
dict	7
other	8

Tabla 3: Identificadores de tipo de datos

1.2 Registro de objetos

A continuación se muestra el formato que tienen el registro de los diferentes objetos dentro del capturador de huellas:

1.2.1 Función

Se describe el registro del objeto función:

eventId	objectId	functionId	functionName	argsN	{argName _i }	argId _i }	fileName
int	int	int	str	int	str	int	str

Tabla 4: Registro del objeto función

1.2.2 Variable local

Se describe el registro del objeto variable local:

eventId	objectId	localId	parentId	localName
int	int	int	int	str

Tabla 5: Registro del objeto variable local

1.2.3 Clase

Se describe el registro del objeto clase:

eventId	objectId	classId	className	classBases
int	int	int	str	¹

Tabla 6: Registro del objeto clase

1.2.4 Método

Se describe el registro del objeto método:

eventId	objectId	methodId	classId	methodName	argsN	{argName _i }	argId _i }	fileName
int	int	int	int	str	int	str	int	str

Tabla 7: Registro del objeto método

1.2.5 Atributo

Se describe el registro del objeto atributo:

eventId	objectId	attributeId	parentId	attributeName
int	int	int	int	str

Tabla 8: Registro del objeto atributo

1.2.6 Thread

Se describe el registro del objeto thread:

eventId	objectId	threadId	sysId
int	int	int	int

Tabla 9: Registro del objeto thread

¹Aún no se toma decisión para poder registrar las super clases que pueda tener la clase registrada.

1.2.7 Probe

Se describe el registro del objeto probe:

eventId	objectId	Id	parentId	currentLasti	currentLineno
int	int	int	int	int	int

Tabla 10: Registro del objeto probe

1.3 Llamada de objetos

A continuación se muestra el formato que tienen las llamadas de los objetos función y método dentro del capturador de huellas:

1.3.1 Función

Se describe la llamada al objeto función:

eventId	objectId	functionId	argsN	{typeId _i }	argValue _i }
int	int	int	int	int	value or valueId ¹

Tabla 11: Llamada al objeto función

1.3.2 Método

Se describe la llamada al objeto método:

eventId	objectId	methodId	targetId	classId	argsN	{typeId _i }	argValue _i }
int	int	int	int	int	int	int	value or valueId ¹

Tabla 12: Llamada al objeto método

Es importante señalar que todas estas llamadas estan acompañadas de los siguientes datos que se describen a continuación:

probeId	parentTimeStampFrame	depth	currentTimeStamp	threadId
int	double	int	double	int

Tabla 13: Coordenadas

¹Aún no se implementa la modificación de la máquina virtual de Python para obtener el identificador único del objeto

1.4 Instanciación de clase

A continuación se muestra el formato que tienen la instanciación de clase dentro del capturador de huellas:

eventId	behaviorId	targetId	argsN	{typeId _i }	argValue _i }
int	int	int	int	int	value or valueId ¹

Tabla 14: Instanciación de clase

Es importante señalar que la instanciación está acompañada de los siguientes datos 13.

1.5 Asignación - Modificación de objetos

A continuación se muestra el formato que tienen las asignaciones — modificaciones de los objetos variable local y atributo dentro del capturador de huellas:

1.5.1 Variable local

Se describe la asignación - modificación al objeto variable local:

eventId	objectId	localId	parentId	typeId	value
int	int	int	int	int	value or valueId ¹

Tabla 15: Registro del objeto variable local

1.5.2 Atributo

Se describe la asignación - modificación al objeto atributo:

eventId	objectId	attributeId	targetId	typeId	value
int	int	int	int	int	value or valueId ¹

Tabla 16: Registro del objeto atributo

Es importante señalar que todas estas asignaciones - modificaciones están acompañadas de los siguientes datos 13.

1.6 Return

A continuación se muestra el formato que tiene el return dentro del capturador de huellas:

Se describe return:

eventId	behaviorId	typeId	value	probeId	hasThrown
int	int	int	value or valueId ¹	int	bool

Tabla 17: Registro de return

Es importante señalar return está acompañado de los siguientes datos 13.

¹Aún no se implementa la modificación de la máquina virtual de Python para obtener el identificador único del objeto