

Indice generale

Prefazione	xii	
Ringraziamenti	xiii	
Introduzione	xv	
A chi è rivolto questo libro	xv	
Come è organizzato questo libro	xv	
Il codice	xvi	
Altre risorse online	xvii	
L'autore	xvii	
Parte I	Primi passi	1
Capitolo 1	Introduzione a WebAssembly	3
Che cos'è WebAssembly?	4	
Asm.js, il precursore di WebAssembly	4	
Da asm.js a MVP	5	
Quali problemi risolve?	5	
Miglioramenti prestazionali	6	
Tempi di avvio più rapidi rispetto a JavaScript	7	
Possibilità di utilizzare linguaggi diversi da JavaScript nel browser	7	
Opportunità di riutilizzo del codice	8	
Come funziona?	8	
Come funzionano i compilatori	9	
Caricamento, compilazione e istanziazione di un modulo	10	
Struttura di un modulo WebAssembly	12	
Preambolo	13	
Sezioni note	13	
Sezioni personalizzate	13	

	Il formato testuale di WebAssembly	13
	In che senso WebAssembly è sicuro?.....	14
	Quali linguaggi possiamo usare per creare un modulo WebAssembly?.....	15
	Dove possiamo usare il nostro modulo?.....	16
	Riepilogo	16
Capitolo 2	I moduli WebAssembly	19
	Le sezioni note	21
	Sezioni personalizzate	24
	Riepilogo	25
Capitolo 3	Il primo modulo WebAssembly.....	27
	Il toolkit Emscripten.....	27
	Moduli WebAssembly	28
	Quando non usare un modulo WebAssembly	30
	Opzioni di output di Emscripten	30
	Compilare codice C o C++ con Emscripten e utilizzare il template HTML	32
	Far sì che Emscripten generi il plumbing code JavaScript	37
	Compilazione di codice C o C++ con il codice JavaScript generato da Emscripten	38
	Creazione di una pagina web HTML di base da utilizzare nei browser.....	40
	Fare in modo che Emscripten generi il solo file WebAssembly.....	43
	Compilazione di codice C o C++ come side module con Emscripten	45
	Caricamento e istanziazione in un browser	47
	Rilevamento delle funzionalità: come verificare se WebAssembly è disponibile	54
	Casi d'uso pratici	56
	Esercizi	56
	Riepilogo	56
Parte II	Lavorare con i moduli	59
Capitolo 4	Riutilizzo della base di codice C++ esistente	61
	Utilizzo di codice C o C++ per creare un modulo con il plumbing Emscripten.....	63
	Le modifiche al codice C++	64
	Compilazione del codice in un modulo WebAssembly	70
	Creazione della pagina web.....	71
	Creazione del codice JavaScript che interagirà con il modulo	73
	Visualizzazione dei risultati	79

Utilizzo di codice C o C++ per creare un modulo senza Emscripten	79
Modifiche al codice C++	80
Compilazione del codice in un modulo WebAssembly	86
Creazione del codice JavaScript che interagirà con il modulo ...	87
Visualizzazione dei risultati	92
Casi d'uso pratici	92
Esercizi	93
Riepilogo	93

Capitolo 5 Creazione di un modulo WebAssembly che richiama codice JavaScript95

Utilizzo di codice C o C++ per creare un modulo con il plumbing Emscripten	98
Modifiche al codice C++	98
Creazione del codice JavaScript da includere nel file JavaScript generato da Emscripten.....	101
Compilazione del codice in un modulo WebAssembly	103
Modifica del codice JavaScript della pagina web	104
Visualizzazione dei risultati	106
Utilizzo di codice C o C++ per creare un modulo senza plumbing Emscripten.....	107
Modifiche al codice C++	109
Compilazione del codice in un modulo WebAssembly	110
Modifica del codice JavaScript che interagirà con il modulo...	111
Visualizzazione dei risultati	114
Casi d'uso pratici	115
Esercizi	115
Riepilogo	115

Capitolo 6 Un modulo che comunica con JavaScript con puntatori a funzione.....117

Utilizzo di codice C o C++ per creare un modulo con il plumbing Emscripten	117
Utilizzo di un puntatore a funzione fornito al modulo dal codice JavaScript	118
Modifica del codice C++	119
Compilazione del codice in un modulo WebAssembly	124
Modifica del codice JavaScript della pagina web	125
Visualizzazione dei risultati	131
Utilizzo di codice C o C++ per creare un modulo senza plumbing Emscripten.....	132
Utilizzo dei puntatori a funzione forniti al modulo dal codice JavaScript	132
Modifiche al codice C++	134

Compilazione del codice in un modulo WebAssembly	135
Modifica del codice JavaScript che interagirà con il modulo...	136
Visualizzazione dei risultati	145
Casi d'uso pratici	146
Esercizi	146
Riepilogo	146

Parte III Argomenti avanzati 147

Capitolo 7 Linking dinamico: le basi149

Linking dinamico: pro e contro	150
Opzioni di linking dinamico	151
Side module e main module	152
Linking dinamico: dlopen	153
Linking dinamico: dynamicLibraries.....	163
Linking dinamico: API JavaScript WebAssembly	168
Ripasso sul linking dinamico.....	175
Casi d'uso pratici	176
Esercizi	176
Riepilogo	177

Capitolo 8 Linking dinamico: l'implementazione179

Creazione dei moduli WebAssembly.....	182
Suddivisione in due file della logica contenuta nel file validate.cpp.....	182
Creazione di un nuovo file C++ per la logica del modulo Place Order	186
Utilizzo di Emscripten per generare i side module WebAssembly	190
Definizione di una funzione JavaScript per gestire un problema di convalida.....	194
Utilizzo di Emscripten per generare il main module WebAssembly	195
Modifica della pagina web.....	196
Modifica del codice JavaScript della pagina web	200
Visualizzazione dei risultati	210
Casi d'uso pratici	211
Esercizi	212
Riepilogo	212

Capitolo 9 Threading: i web worker e i pthread213

I vantaggi dei web worker.....	214
Considerazioni sull'utilizzo dei web worker	215

Prefetching di un modulo WebAssembly utilizzando un web worker	216
Modifica della logica di <code>calculate_primes</code>	218
Utilizzo di Emscripten per generare i file WebAssembly	220
Copia dei file nella posizione corretta	221
Creazione del file HTML per la pagina web	222
Creazione del file JavaScript per la pagina web	222
Creazione del file JavaScript del web worker	225
Visualizzazione dei risultati	226
Utilizzo dei <code>pthread</code>	226
Modifica della logica di <code>calculate_primes</code> per creare e utilizzare quattro <code>pthread</code>	228
Utilizzo di Emscripten per generare i file WebAssembly	232
Visualizzazione dei risultati	234
Casi d'uso pratici	235
Esercizi	236
Riepilogo	237
Capitolo 10 Moduli WebAssembly in Node.js	239
Ripasso generale	240
Convalida server-side	241
Lavorare con i moduli creati da Emscripten.....	241
Caricamento di un modulo WebAssembly.....	242
Chiamata delle funzioni nel modulo WebAssembly	243
Chiamate nel codice JavaScript	247
Chiamata di puntatori a funzione JavaScript.....	250
Utilizzo dell'API JavaScript WebAssembly	252
Caricamento e istanziazione di un modulo WebAssembly.....	252
Funzioni di chiamata nel modulo WebAssembly.....	254
Il modulo WebAssembly che richiama codice JavaScript.....	258
Il modulo WebAssembly che richiama puntatori a funzione JavaScript.....	262
Casi d'uso pratici	266
Esercizi	267
Riepilogo	267
Parte IV Debug e test	269
Capitolo 11 Il formato testuale di WebAssembly	271
Creazione della logica di base del gioco utilizzando il formato testuale WebAssembly.....	273
Le sezioni del modulo.....	275
Commenti.....	277
Signature di funzione.....	277
Il nodo <code>module</code>	278

I nodi import.....	279
I nodi global.....	283
I nodi export.....	285
Il nodo start.....	286
I nodi code.....	287
I nodi type.....	309
Il nodo data.....	311
Generazione di un modulo WebAssembly dal codice in formato testuale.....	312
Il modulo generato da Emscripten.....	312
Creazione del file C++.....	312
Generazione di un modulo WebAssembly.....	315
Creazione dei file HTML e JavaScript.....	315
Modifica del file HTML.....	316
Creazione del file JavaScript.....	318
Visualizzazione dei risultati.....	324
Casi d'uso pratici.....	324
Esercizi.....	325
Riepilogo.....	325

Capitolo 12 Debug327

Estendere il gioco.....	328
Modifica del codice HTML.....	330
Visualizzazione del numero di tentativi.....	330
La funzione JavaScript generateCards.....	331
Modifica del codice in formato testuale.....	333
Generazione del file Wasm.....	333
Test delle modifiche.....	335
Incremento del numero di tentativi.....	337
La funzione JavaScript updateTriesTotal.....	338
Modifica del codice in formato testuale.....	339
Generazione del file Wasm.....	340
Verifica delle modifiche.....	342
Aggiornamento della schermata di riepilogo.....	349
La funzione JavaScript levelComplete.....	350
Modifica del codice in formato testuale.....	351
Generazione del file Wasm.....	352
Verifica delle modifiche.....	352
Esercizi.....	354
Riepilogo.....	354

Capitolo 13 Test, e poi cosa?.....357

Installazione del framework di test JavaScript.....	359
Il file package.json.....	360
Installazione di Mocha e Chai.....	360

Creazione ed esecuzione di test.....	361
Scrittura dei test.....	361
Esecuzione dei test dalla riga di comando.....	366
Una pagina HTML che carica i vostri test	366
Esecuzione dei test da un browser.....	368
Superamento dei test	370
E ora?.....	370
Esercizi	372
Riepilogo	372

Appendice A Installazione e configurazione degli strumenti373

Python	373
Esecuzione di un server web locale	374
Il media type WebAssembly.....	375
Emscripten	376
Download dell'SDK Emscripten	377
Se state usando un sistema Windows	379
Se state usando un sistema Mac o Linux.....	379
Risolvere i problemi di installazione.....	379
Node.js.....	380
WebAssembly Binary Toolkit.....	381
Bootstrap	381

Appendice B ccall, cwrap e chiamate dirette alle funzioni383

ccall.....	383
Costruzione di un semplice modulo WebAssembly.....	384
Costruzione della pagina web che dialogherà con il modulo WebAssembly.....	385
cwrap	386
Modifica del codice JavaScript per utilizzare cwrap	386
Chiamate dirette di funzioni	387
Passaggio di un array a un modulo	388

Appendice C Le macro in Emscripten391

Le macro emscripten_run_script.....	391
Le macro EM_JS	392
Nessun parametro	393
Passaggio di valori per i parametri.....	394
Passaggio di parametri puntatori	395
Restituzione di un puntatore a una stringa.....	397
Le macro EM_ASM	399
EM_ASM.....	400
EM_ASM_.....	401
Passaggio di parametri che sono puntatori.....	401
EM_ASM_INT e EM_ASM_DOUBLE	403
Restituzione di un puntatore a una stringa.....	404

Appendice D Soluzioni degli esercizi.....407

Appendice E Approfondimenti sul formato testuale.....431

Istruzioni di controllo del flusso	432
Le istruzioni if	432
I cicli	437
I puntatori a funzione	445
Sottoporre a test il codice	447

Indice analitico.....449