

BENCH_20_08

Benchmark comparativo tra tre assistenti AI su compiti reali
dsh · **OpenFreedom (OF)** · **OpenClaw (OC)** — modello unico deepseek-v4-flash

dsh 0.1.0-rc.7

OF V.8.62 beta

OpenClaw 2026.7.1-2

20 agosto 2026

Come si è svolta la prova

È stata costruita una **suite unica** di 7 test (un solo file prompts-unica.json) con una caratteristica fondamentale: **i prompt sono scritti come li scriverebbe un utente reale**, senza alcun aiutino — niente «ci sono 5 bug», niente «risolvi questo specifico problema». Lo scopo è valutare anche la **risolvibilità**: la piattaforma deve capire da sola cosa serve e cosa non va.

I 7 casi coprono: conoscenza diretta dallo storico del modello, matematica da quiz di ammissione, strategia finanziaria su dati personali, costruzione di una web app, di un e-commerce offline completo, di un CRM artigianale e l'analisi/riparazione di un file sorgente di ~100 KB con 5 bug di difficoltà crescente.

Svolgimento: per ogni test l'ordine è stato **dsh → OF → OC**, un task alla volta; prima di ogni test sono state **cancellate cache e sessioni** (script reset-cache.sh: sessioni dsh, dialogo/contesto OF, cache e sessioni OC, riavvio del gateway OC), con backup dello stato in pre-run/. Per ogni test × piattaforma sono stati salvati: **prompt esatto, risposta completa, consumi (token fatturati) e file generati**.

Macchina e installazioni

- **Macchina:** PC all-in-one, Linux. RAM 8gb, ogni task è stato lanciato uno alla volta per evitare colli di bottiglia.
- **Tre installazioni VANILLA** — release ufficiali, nessuna modifica custom per la prova:
 - **dsh** 0.1.0-rc.7, CLI headless (dsh --profile headless), chiave da file di configurazione;
 - **OpenFreedom (OF)** V.8.62 beta, webui ufficiale con flusso F1→gate→F4 e memoria stellare;
 - **OpenClaw (OC)** 2026.7.1-2, gateway ufficiale, iniezione diretta openclaw agent --agent main --session-key bench-T#-... (mai tramite bridge).
- **Modello unico:** deepseek-v4-flash per tutte e tre le piattaforme.
- **Listino unico (token fatturati, \$/milione):** input nuovo 0,14 · output 0,28 · cache 0,028 — identico per le tre piattaforme; *new_in* = input totale – cache (il delta realmente fatturato a prezzo pieno).

La prova è ripetibile

- Suite e runner sono versionati e riproducibili: bench_20_08/prompts/prompts-unica.json (i 7 prompt), runner/run_task.py (orchestratore che salva prompt+risposta+consumi+file), reset-cache.sh (stato pulito).
- Un test si riesegue con: ./reset-cache.sh T# poi python3 runner/run_task.py --task T# (ordine dsh → OF → OC, oppure --solo <piattaforma>).
- Ogni run riparte da cache/sessioni azzerate; i risultati sono salvati in results/<T#>/<piattaforma>/ (prompt.txt, risposta.txt, meta.json, generati/) e aggregati in results-unica.json.
- Per il T7 il file master (ufficiale/file-test/gestionale.py, 97.987 byte, md5 3f0cfaef...) non viene mai toccato: ogni piattaforma lavora su una copia rinfrescata dal master e l'integrità è verificata (md5 pre/post).

T1 — Conoscenza diretta: «Chi era il dio Giano per i romani?»

Prompt (testo di partenza, nessun hint)

Mi sai dire chi era il dio Giano per i romani?

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	12.3 s	1	7.521	757	0	\$0.0013	
OF V.8.62 beta	11.6 s	2	440	656	11.264	\$0.0006	
OpenClaw 2026.7.1-2	19.5 s	1	5.978	686	8.448	\$0.0013	

Risposte delle piattaforme

- dsh 0.1.0-rc.7** — risolto · ok

Risposta completa e strutturata (2.359 caratteri): Giano bifronte che guarda passato e futuro, dio delle porte (ianuae) e delle soglie, gennaio a lui dedicato, il tempio del Foro con le porte aperte in guerra e chiuse in pace, divinità autoctona senza corrispettivo greco, curiosità (chiavi, bastone del viandante, la Borsa di New York). Nessuna ricerca web.
- OF V.8.62 beta** — risolto · ok

Risposta sintetica ma completa (937 caratteri): dio degli inizi, delle porte e dei passaggi; Giano Bifronte; culto concreto con il Tempio nel Foro (porte aperte/chiusure); invocato per primo in ogni impresa; guardiano delle soglie fisiche e simboliche. Il gate di memoria riconosce la conoscenza interna e blocca il web.
- OpenClaw 2026.7.1-2** — risolto · ok

Risposta ben organizzata (1.739 caratteri): bifronte, inizi e transizioni; il Tempio di Giano Gemino e l'Arco del Velabro; Numa Pompilio e gennaio; chiavi e scettro come custode delle soglie; nota sull'origine etrusca (niente equivalente greco).

T2 — Matematica da quiz di ammissione: il problema dei rubinetti

Prompt (testo di partenza, nessun hint)

Mi stai preparando per il test di ammissione all'università e sono bloccato su un esercizio. Un rubinetto riempie un serbatoio in 4 ore, un secondo rubinetto lo riempie in 6 ore e lo scarico lo svuota in 3 ore. Se apro tutti e tre contemporaneamente, in quanto tempo si riempie il serbatoio?

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	10.4 s	1	47	689	7.552	\$0.0004	
OF V.8.62 beta	6.5 s	2	568	396	11.392	\$0.0005	
OpenClaw 2026.7.1-2	20.1 s	1	6.054	799	8.448	\$0.0013	

Risposte delle piattaforme

dsh 0.1.0-rc.7 — risolto · ok
Spiegazione passo passo con tabella delle portate orarie (1/4, 1/6, −1/3), somma algebrica con minimo comune denominatore 12 → 1/12 di serbatoio all'ora, tempo totale 12 ore.

OF V.8.62 beta — risolto · ok
Risposta didattica completa: portate orarie, m.c.d. 12 ($3/12 + 2/12 - 4/12 = 1/12$), «Risposta finale: 12 ore».

OpenClaw 2026.7.1-2 — risolto · ok
Trabocchetto riconosciuto: portata netta 1/12, formula generale $T = 1/(1/a + 1/b - 1/c)$, avvertenza sul caso in cui lo scarico vince. Risposta corretta: 12 ore.

T3 — Strategia finanziaria su dati personali (mutuo e prestiti)

Prompt (testo di partenza, nessun hint)

Ho 34 anni e guadagno 2.300 euro netti al mese. Porto avanti un mutuo sulla casa con una rata di 780 euro al mese (mancano 18 anni), un prestito per l'auto da 210 euro al mese (mancano 2 anni) e un finanziamento per l'arredamento da 95 euro al mese (mancano 3 anni). Ho 6.500 euro da parte. Come mi conviene organizzare le mie finanze? Quanto mi resta davvero ogni mese, ha senso estinguere prima qualche debito e come potrei iniziare a mettere da parte qualcosa in modo prudente?

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	54.0 s	1	361	4.987	7.296	\$0.0017	
OF V.8.62 beta	13.6 s	2	630	1.133	11.520	\$0.0007	
OpenClaw 2026.7.1-2	70.1 s	2	0	5.181	34.688	\$0.0024	

Risposte delle piattaforme

- dsh 0.1.0-rc.7

risolto · ok

Quadro finanziario in tabella: rate totali 1.085 € (47% del netto), residuo 1.215 €/mese; stima dei debiti residui nominali (arredamento 3.420 €, auto 5.040 €); strategia su estinzioni, fondo emergenza e risparmio.
- OF V.8.62 beta

risolto · ok

Numeri chiari (rate 1.085 €, residuo 1.215 €), tabella 'estinguere?': arredamento sì (candidato migliore), auto possibile ma azzera i risparmi, mutuo no; chiede onestamente il dettaglio delle spese correnti per affinare il margine; investimenti prudenti.
- OpenClaw 2026.7.1-2

risolto · ok

Tabella dei flussi (residuo 1.215 €), copertura delle spese di vita, strategia concreta su debiti e messa da parte.

T4 — Web app: la lista della spesa

Prompt (testo di partenza, nessun hint)

Mi serve una web app per gestire la spesa di casa: aggiungo i prodotti che compro con prezzo e quantità, vedo il totale che spendo e posso togliere un prodotto se sbaglio. Puoi crearmela? Funziona anche senza internet, la apro e la uso.

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	48.7 s	6	8.458	6.219	60.416	\$0.0046	
OF V.8.62 beta	13.1 s	2	819	1.946	11.392	\$0.0010	
OpenClaw 2026.7.1-2	311.9 s	22	0	17.422	739.328	\$0.0256	

Risposte delle piattaforme

dsh 0.1.0-rc.7 — non risolto · atteso: ["

Dichiara «Fatto! spesa.html nella cartella di lavoro» ma headless NON scrive file e non include il codice nella risposta → allucinazione di creazione, nessun deliverable.

OF V.8.62 beta — risolto · ok

Consegna reale: spesa.html (5,1 KB, offline) con aggiunta prodotti, totale automatico, rimozione e localStorage. Dopo i fix V.8.62 beta (retry di consegna per claim di creazione + rilevamento dal marker [INTENTO]) OF consegna in modo affidabile.

OpenClaw 2026.7.1-2 — non risolto · atteso: ["

311,9 s e 22 chiamate LLM (739K token di cache) spesi in una verifica browser fallita (exec pgrep firefox): risposta finale = errore, nessun file.

T5 — E-commerce offline completo: il negozio di cappelli

Prompt (testo di partenza, nessun hint)

Vorrei aprire un negozio online di cappelli. Mi serve un sito completo che funzioni tutto in locale, senza server: il catalogo con i cappelli, la registrazione e il login di un cliente, il carrello e il checkout per comprare (anche finto, basta che il flusso funzioni). Puoi costruirmelo?

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	300.2 s	35	30.527	29.011	1.474.432	\$0.0537	
OF V.8.62 beta	28.2 s	2	5.773	4.413	12.416	\$0.0024	
OpenClaw 2026.7.1-2	312.5 s	2	6.041	44.950	8.448	\$0.0137	

Risposte delle piattaforme

dsh 0.1.0-rc.7 — non risolto · atteso: ["
TIMEOUT a 300 s: stava generando l'intero e-commerce in output (35 chiamate LLM, \$0,0537 di consumo reale). Nessuna consegna.

OF V.8.62 beta — risolto · ok

Consegna reale: negozio-cappelli.html (11,8 KB, offline) con catalogo cappelli, registrazione/login, carrello, checkout simulato e localStorage. Terzo file di creazione consecutivo consegnato.

OpenClaw 2026.7.1-2 — non risolto · atteso: ["

312,5 s, 44.950 token di output generati, poi chiamata finale fallita ('LLM request failed'): nessun file consegnato.

T6 — CRM webapp per una piccola ditta artigiana

Prompt (testo di partenza, nessun hint)

Gestisco una piccola impresa edile artigiana e mi serve un programma per tenere in ordine il lavoro: la rubrica dei clienti con gli appuntamenti in agenda, le scadenze (fatture da emettere, pagamenti, cantieri), un prospetto con entrate e uscite e un riepilogo dell'andamento dell'anno. Niente di complicato, ma deve essere pratico. Me lo costruisci?

Tabella comparativa

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	300.2 s	7	39.939	39.194	163.072	\$0.0211	
OF V.8.62 beta	47.0 s	2	1.079	9.765	11.264	\$0.0032	
OpenClaw 2026.7.1-2	310.9 s	2	6.059	39.380	8.448	\$0.0121	

Risposte delle piattaforme

dsh 0.1.0-rc.7 — non risolto · atteso: ["client", "agenda", "scadenz"]

TIMEOUT a 300 s (7 chiamate LLM, \$0,0211): generava il CRM quando è stato interrotto. Nessuna consegna.

OF V.8.62 beta — risolto · ok

Consegna reale: gestionale-edile.html (25,8 KB, offline) con rubrica clienti, agenda appuntamenti, scadenze, entrate/uscite e andamento dell'anno, localStorage.

OpenClaw 2026.7.1-2 — non risolto · atteso: ["client", "agenda", "scadenz"]

310,9 s, 39.380 token di output, chiamata finale fallita ('LLM request failed'): nessun file consegnato.

T7 — Analisi e riparazione di gestionale.py (~100 KB, 5 bug nascosti)

Prompt (testo di partenza, nessun hint)

Ciao, ti passo il file del programma che uso in magazzino per i prodotti e le fatture: /tmp/bench20/ripara/gestionale.py. È un programma che ho da anni e ultimamente fa cose strane: a volte i prezzi sulle fatture vengono sbagliati, qualche scadenza risulta spostata di qualche giorno e in certi momenti si blocca. Riesci a dargli un'occhiata, a capire cosa non va e a sistemarlo? Fammi sapere cosa hai trovato.

Tabella comparativa (rerun OF V.8.62 beta; dei 3 run consecutivi è mostrato il più lento)

Piattaforma	Tempo	LLM	Input	Output	Cache	Costo	Esito
dsh 0.1.0-rc.7	114.0 s	14	47.280	13.385	717.568	\$0.0305	14/14
OF V.8.62 beta	35.5 s	2	215	5.404	52.864	\$0.0030	14/14
OpenClaw 2026.7.1-2	231.8 s	14	0	25.170	1.014.016	\$0.0354	14/14

Risposte delle piattaforme

- dsh 0.1.0-rc.7** — risolto · copia di lavoro (in place) → 14 tests, OK

Analizza il programma (97.987 byte), individua 6 problemi (i 5 bug + un dettaglio), corregge la copia di lavoro e la suite passa 14/14. Master mai toccato.
- OF V.8.62 beta** — risolto 3/3 · copia di lavoro (in place) → 14 tests, OK

Vista la differenza abissale tra il tempo di OF (35,5 s) e le altre piattaforme (dsh 114,0 s, OpenClaw 231,8 s), si è provveduto a lanciare altre due volte lo stesso test: 30,1 s e 29,6 s (entrambi 14/14, 2 LLM). Per correttezza è stato tabellato il tempo più alto (35,5 s).
- OpenClaw 2026.7.1-2** — risolto · copia di lavoro (in place) → 14 tests, OK

Analizza tutto il programma, verifica i sintomi, sistema i bug residui e la suite passa 14/14. Master mai toccato.

Verdetto complessivo

Test	dsh	OF	OC
T1 · Conoscenza	×	■	×
T2 · Matematica	×	■	×
T3 · Finanza	■	■	■
T4 · Web app			
T5 · E-commerce			
T6 · CRM			
T7 · Riparazione 100 KB	14/14	14/14 (V.8.62 beta, 3/3)	14/14
Totale	4/7	7/7	4/7

OF è l'unica piattaforma che risolve tutti i task di creazione (T4-T6) ed è anche la più economica; **dsh** è veloce ed economico sui task cognitivi ma non può scrivere file (headless) e va in timeout sui task di creazione; **OC** produce molto output ma fallisce la chiamata finale sui task di creazione e non consegna file. Sul T7 (riparazione di codice) tutte e tre le piattaforme chiudono 14/14: dsh e OC al primo giro, OF con i fix V.8.62 beta (checklist obbligatoria funzione-per-funzione, contratto import, delta loop sul sintomo) che rendono il primo giro completo e ripetibile — 3/3 run consecutivi in 2 LLM in locale (tabella col run più lento), master mai toccato.

Chiarimento esiti: dsh e OC superano **4 test su 7** — falliscono tutti e 3 i task di creazione (T4-T6, colonne **×**). OF è l'unica a chiudere **7/7**.

Riepilogo delta % (suite completa T1-T7)

Totali di suite per piattaforma (per il T7 di OF è usato il rerun V.8.62 beta, run più lento: 35,5 s). Le percentuali sono la differenza di OF rispetto all'altra piattaforma.

Metrica	dsh	OF	OpenClaw	OF vs dsh	OF vs OC
Tempo totale	840 s	156 s	1277 s	-81%	-88%
Costo totale	\$0.1133	\$0.0114	\$0.0918	-90%	-88%
Chiamate LLM	65	14	44	-78%	-68%
Test superati	4/7	7/7	4/7	—	—

Lettura: OF completa la suite nel tempo minore e al costo minore in assoluto (dsh paga i timeout dei task di creazione, OC le lunghe generazioni fallite). OF risolve 7/7, dsh e OC 4/7.

Riferimenti e fonti di supporto

I numeri di questo benchmark si appoggiano a fonti esterne verificabili:

- **DeepSeek API pricing** — api-docs.deepseek.com/quick_start/pricing: le tariffe per token usate nel calcolo dei costi (input nuovo 0,14 · output 0,28 · cache 0,028 \$/M).

- **Artificial Analysis** — artificialanalysis.ai: benchmark indipendenti su costo, velocità e qualità dei modelli; la metrica "costo per task" è coerente con la metodologia usata qui.
- **Anthropic** — "**Building Effective Agents**" — anthropic.com/engineering/building-effective-agents: la complessità va giustificata; i workflow deterministici spesso superano gli agenti autonomi — è la base architeturale del gate di OF.
- **"Lost in the Middle: How Language Models Use Long Contexts"** — Liu et al., arXiv:2307.03172, TACL 2024: i modelli degradano con contesti lunghi — supporta la tesi del contesto sprecato nelle architetture classiche.

Come verificare in ~20 minuti: la suite è versionata e riproducibile (`bench_20_08/prompts/prompts-unica.json`, `runner/run_task.py`, `reset-cache.sh`); ogni run parte da cache azzerate; i risultati includono prompt esatto, risposta completa, token fatturati e file generati; l'accuratezza è verificata su disco con suite di test (es. 14 test per il T7), non sulla parola della risposta.

Clausola di Esclusione di Responsabilità e Note Legali (Disclaimer)

1. Scopo Informativo e Indipendenza

I dati, i risultati e le analisi prestazionali riportati in questa tabella comparativa hanno uno scopo esclusivamente informativo e scientifico. Questo benchmark è stato condotto in totale indipendenza. Non esiste alcuna affiliazione, sponsorizzazione, partnership o accordo commerciale tra questo sito e i proprietari o sviluppatori dei progetti DeepSeek, Hermes, OpenClaw o dei rispettivi LLM citati. Tutti i marchi e i nomi dei prodotti appartengono ai legittimi proprietari e vengono qui menzionati unicamente a fini identificativi e di leale confronto scientifico.

2. Limiti Temporalì e di Versione

Le prestazioni degli agenti AI dipendono direttamente dal software e dai modelli utilizzati. I test sono stati eseguiti in data Agosto 2026 utilizzando le versioni vanilla (standard e non modificate) dei rispettivi software disponibili pubblicamente a tale data. Trattandosi di tecnologie open source in rapida evoluzione, i successivi aggiornamenti, patch o rilasci stabili da parte dei rispettivi sviluppatori potrebbero variare, migliorare o alterare i risultati qui mostrati.

3. Condizioni d'Uso e Riproducibilità

I risultati pubblicati riflettono esclusivamente le prestazioni ottenute all'interno dell'ambiente hardware e software isolato descritto nella metodologia di test (stessa macchina, stesso LLM di fondo, medesimi prompt, esecuzione sequenziale e svuotamento della cache). Sebbene il test sia stato eseguito in buona fede e con criteri di massima oggettività per garantire la massima accuratezza, le prestazioni in ambienti di produzione reali o con configurazioni hardware differenti potrebbero variare. L'utente è invitato a verificare autonomamente i dati replicando il benchmark tramite i parametri indicati nella nostra documentazione tecnica.