

CalcoloVoti

Gruppo	MAKALL	MAKINS	MAKCLIN	Qualità Maketile	Average	SRV.REX	MON.REX	MON.AFOK	MON.AFER	MON.CHK	MON.CPT	MON.BF	Average	LIB.AFOK	LIB.AFER	LIB.CPT	LIB.PG	LIB.CAL	LIB.BEA	Average	SMV.PER	SMV.PRO	Average	Qualità codice	Commenti	Molularità	Average	Struttura e informazioni Site e testo e grafica	Relazione	Average	Media pesata	Voto base	Bonus gruppo 1p-2pt.	Estensioni opzionali Max 2 punti	penalizzazioni per errori nel codice	Voto progetto	Voto es. 1	NOTE
Compito-Perfetto	10	10	10	10	10.0	10	10	10	10	10	10	10	10.0	10	10	10	10	10	10	10.0	10	10	10.0	10	10	10	10	10	10	10.0	33.0	0	0	0	33.00	33		
Di Mauro	5	5	8	7	6.3	8	9	9	10	10	5	10	8.7	9	10	n/a	n/a	n/a	n/a	9.5	n/a	n/a	###	5	6	7	6.0	8	8	8.0	#DIV/0!	1		3	#DIV/0!		ORALE	
Duse	5	5	8	7	6.3	8	9	9	10	10	5	10	8.7	9	10	n/a	n/a	n/a	n/a	9.5	n/a	n/a	###	5	6	7	6.0	8	8	8.0	#DIV/0!	1		3	#DIV/0!		ORALE	
Gambero	10	8	9	9	9.0	10	10	10	10	10	10	10	10.0	10	10	10	10	10	10	10.0	10	10	10.0	9	8	10	9.0	10	7	8.5	31.4	2			33.43	30		
Santucci	6	8	8	7	7.3	6	6	n/a	n/a	n/a	n/a	n/a	6.0	10	7	7	5	10	5	7.3	3	10	6.5	6	4	8	6.0	5	5	5.0	20.9	1		2	19.91		ORALE	
Parisse	6	8	8	7	7.3	6	6	n/a	n/a	n/a	n/a	n/a	6.0	10	7	7	5	10	5	7.3	3	10	6.5	6	4	8	6.0	5	5	5.0	20.9	1		2	19.91		ORALE	
Sciacca	10	8	9	9	9.0	10	10	8	8	10	0	10	8.0	10	10	0	10	10	10	8.3	10	10	10.0	9	1	8	6.0	8	8	8.0	25.9	0		2	23.91		ORALE	
Goythom	10	8	9	9	9.0	10	10	8	8	10	0	10	8.0	10	10	0	10	10	10	8.3	10	10	10.0	9	1	8	6.0	8	8	8.0	25.9	0		2	23.91		ORALE	
de Fabritis	10	8	9	9	9.0	10	10	8	8	10	0	10	8.0	10	10	0	10	10	10	8.3	10	10	10.0	9	1	8	6.0	8	8	8.0	25.9	0		2	23.91		ORALE	
Greco	10	9	10	9	9.5	10	10	10	10	10	10	10	10.0	10	10	10	10	10	10	10.0	10	10	10.0	9	9	9	9.0	9	10	9.5	32.0	2			34.01		ORALE	
Di Rollo	10	8	10	9	9.3	10	9	9	9	9	9	9	9.1	10	10	7	7	10	7	8.5	0	10	5.0	7	4	7	6.0	9	10	9.5	27.0	0			26.97	22		
Codastefano	10	8	10	9	9.3	10	9	9	9	9	9	9	9.1	10	10	7	7	10	7	8.5	0	10	5.0	7	4	7	6.0	9	10	9.5	27.0	0			26.97		ORALE	
Yu	10	8	10	9	9.3	10	9	9	9	9	9	9	9.1	10	10	7	7	10	7	8.5	0	10	5.0	7	4	7	6.0	9	10	9.5	27.0	0			26.97	27		
Sforza	7	8	8	7	7.5	10	4	n/a	n/a	n/a	n/a	n/a	7.0	n/a	n/a	n/a	n/a	n/a	n/a	###	n/a	n/a	###	7	7	7	7.0	9	9	9.0	#DIV/0!	0			#DIV/0!		ORALE	
Moauo	7	8	8	7	7.5	10	4	n/a	n/a	n/a	n/a	n/a	7.0	n/a	n/a	n/a	n/a	n/a	n/a	###	n/a	n/a	###	7	7	7	7.0	9	9	9.0	#DIV/0!	0			#DIV/0!	20		
Arcadu	7	8	8	7	7.5	10	4	n/a	n/a	n/a	n/a	n/a	7.0	n/a	n/a	n/a	n/a	n/a	n/a	###	n/a	n/a	###	7	7	7	7.0	9	9	9.0	#DIV/0!	0			#DIV/0!		ORALE	

COMpito-Perfetto 33

Ritardo.
nsuffic.
Ritardo.
nsuffic.
Ritardo.
Ritardo.
Ritardo.
Ritardo.
nsuffic.
nsuffic.
nsuffic.

Makefile	make ricompila sempre tutto. Make install non sovrascrive in caso di vecchia installazione. Make install non funziona se prima non è stato fatto make all. Make clean ritorna con errori se lanciato più volte
SRV.REX	NO non sono create fifo ma file regolari (molto grave!)
MON.REX	NO dopo stop smmd resta attivo come zombie!
MON.AFOK	NO non ci sono messaggi di ritorno da parte del monitor
MON.AFER	NO free senza parametri causa Bad address
MON.CPT	n/a
MON.BF	n/a
LIB.AFOK	n/a Libreria non riesce a comunicare con il server.
LIB.CPT	n/a
LIB.CAL	n/a
LIB.REA	n/a
Qualità codice	Stile costante e leggibile. Funzioni di lunghezza abbastanza uniforme. Buon uso di macro Poco elegante l'uso di un vettore statico piuttosto che quello di una lista linkata per la memoria. Errore nella gestione dei processi (zombie) e delle fifo (assenti). Assenza dell'uso della funzione memcpy (cfr. Funzioni get/put) Scelta del protocollo di comunicazione da illustrare.
Modularità	Migliorabile. Buona anche se manca un po' in chiarezza. Diverse frasi sgrammaticate. Non risulta chiaro il meccanismo di assegnazione degli indirizzi logici.
Relazione	Migliorabile la descrizione del protocollo di comunicazione tra i processi.
Commenti	Presenti ma migliorabili
NOTE	Non sufficiente a causa della mancanza dell'uso di fifo. Libreria non funziona

Makefile	make install rimuove tutta la directory e riscrive sempre tutto (non ok)
SRV.REX	SI non sono create fifo ma file regolari (molto grave!)
MON.REX	SI
MON.AFOK	OK Ma free vuole i puntatori diversi da come li riportano alloc e lab Molti messaggi di debug!
MON.AFER	OK
MON.CPT	OK Puntatori diversi da quelli riportati nei messaggi del monitor.
MON.BF	OK
LIB.AFOK	OK
LIB.CPT	NO 1 errore la prima esecuzione, 3 la seconda, 6 le successive.
LIB.CAL	OK
LIB.REA	NO 3 errori e 25 successi, ogni esecuzione lascia in memoria dei blocchi anormali
SMM.PER	NO Impossibile da testare a causa della scelta della modalità di assegnamento dei puntatori logici
	Stile costante e leggibile. Scarso l'uso di macro (comunque migliorabile) Inutile allocare buffer per put e get Non molto comodo l'uso dell'allocazione dinamica per lo scambio di messaggi. Esiti delle malloc non controllati Chiedere execl La lunghezza di alcune funzioni sono sbilanciate rispetto alle altre
Qualità codice	Molto codice ripetuto che poteva essere inserito in funzioni
Commenti	Quasi assenti e comunque non nei posti importanti
Modularità	Migliorabile Chiedere il significato di: "La libreria è di tipo statico, ciò significa che viene linkata direttamente all'eseguibile, senza i ricompilare il codice."
Relazione	Non si capisce cosa succede quando il puntatore logico diventa talmente alto da non essere più gestibile (cfr test SMM.I)

NOTE

Makefile	make install usa l'utility install. L'opzione -C non viene riconosciuta
SRV.REX	NO smmd avviato senza parametri non va (fuori spec?)
MON.REX	NO start senza parametri è indicato come comando errato: manca un messaggio di errore più esaustivo.
MON.AFOK	OK
MON.AFER	OK Gli errori non sono indicati in maniera adeguata (genericamente Memory allocated at address: (nil))
MON.CPT	OK
MON.BF	OK
LIB.AFOK	OK
LIB.CPT	OK
LIB.CAL	OK
LIB.REA	OK
Qualità codice	Stile chiaro e lineare. Codice molto leggibile. Unica pecca: payload allocato dinamicamente. Possibile evitarlo? Buddy system altera la dimensione del blocco richiesto, adattandola alla più piccola potenza di due (fuori spec?)
Modularità	Buona
Relazione	Molto buona. Non risulta chiaro il meccanismo di assegnazione degli indirizzi logici: cosa vuol dire "gestendo gli spazi di indirizzamento separatamente per ogni processo"? Vuol dire che gli indirizzi logici sono gli stessi per ogni processo?
Commenti	Buoni ma migliorabili
NOTE	

```

./src/smmMon.c:85: warning: implicit declaration of function 'kill'
Make uninstall rimuove la cartella smm. Perché?
Makefile      Make install ricopia sempre tutto a prescindere.

SRV.REX      OK
MON.REX      NO          Molto grave: execl invocata con path relativo non permette l'avvio da ~/smm
lab senza parametri non funziona (fuori specifica)
MON.AFOK     NO          Con buddy system, lab non riporta i blocchi allocati e lfb non riporta i blocchi lit
MON.AFER     NO          alloc/lab/free senza parametri non funzionano (non vengono accettati come co
MON.CHK      OK
MON.CPT      NO          Il comando che dovrebbe innescare la compattazione restituisce "error"
MON.BF       OK
LIB.CPT      NO          Non sembra funzionare.

Errore grave nella execl.
Stile chiaro e preciso.
Manca compattazione.
Qualità codice      Buon uso di macro.
Commenti            Assenti
Modularità          Buona.
Relazione           Ben articolata, ma mancano informazioni sugli indirizzi logici/fisici e sulla mancanza dei
Estensione          buddy system

```

	serie warning: assignment makes pointer from integer without a cast. serie warning: format '%x' expects type 'unsigned int', but argument 2 has type 'void *'. serie warning: unused variable in funzioni_buddy.c smmd.c warning: return type of 'main' is not 'int' avviaServer.c warning: implicit declaration of function 'waitpid'
Makefile	
SRV.REX	Smmd non si avvia senza parametri: genera segmentation fault e lascia fifo e file temporanei nel filesystem
MON.REX	smmMon avvia il server ma non torna alla shell: il processo padre è bloccato sulla waitpid. Perché?
MON.AFOK	È impossibile interagire con la shell. Dopo l'avvio di smmd e di smmMon, alloc non funziona e genera segmentation fault in smmd
MON.CPT	Impossibile testare
MON.BF	Impossibile testare
LIB.*	LIB.CAL provoca errori se lanciato in batch. Non se lanciato da solo
LIB.CPT e REA	Non funziona bene: se lanciato più volte genera errori. Include con path relativo Scorso uso di macro Non chiaro perché se la fifo è già presente il server non si debba avviare. Non controllato il numero di argomenti del main (molto grave). Put e get fanno un uso inutile del buffer. Non chiaro perché il monitor si metta in attesa su una waitpid... (molto grave). Perché la shell legge solo 30 caratteri?
Qualità codice	Non chiara la logica del main di smmMon
Commenti	Scarsi
Modularità	Buona
Relazione	Insufficiente: mancano informazioni sugli indirizzi logici/fisici, sulla procedura di compattamento, sul protocollo di comunicazione (giusto per citarne qualcun)
Estensione	buddy system

Makefile	make install può essere eseguito 2 volte... perché? make uninstall rimuove anche la cartella smm... not ok
SRV.REX	OK
MON.REX	OK
MON.AFOK	OK
MON.CPT	OK
MON.BF	OK
LIB.AFOK	OK
LIB.CPT	OK
Qualità codice	Stile costante e ordinato. Un po' contorto in alcuni punti: poca leggibilità del codice. -Funzioni di lock "scomode", perché usano due descrittori. Per migliorare la leggibilità è preferibile chiamare due volte la stessa funzione su un singolo descrittori -smm_kfifo.c l180: se la seconda fcntl fallisce, il primo lock non viene rilasciato -smmd_lib.c l785: perché allocare buffer se poi si copia in una zona già allocata? -smmMon_lib.c l214: chiedere argomenti exec l
Commenti	Adeguati
Modularità	Molto buona -Inglese con molti errori! ;-) -Sbagliato che il lock su un file già bloccato genera errore (The routine kfifo lock() invokes the system call fcntl() asking for a write lock on the entire FIFO file. If another process keeps a lock then the above routine returns with error).
Relazione	
NOTE	

		Make ricompila tutto ogni volta. Perché? Come si può modificare? Manca make uninstall. Make clean discutibile.
		Make esegue anche le funzioni di make install... Perché?
Makefile		Impossibile compilare smm_test (modifiche necessarie...)
SRV.REX	NO	smmd avviato senza parametri genera segmentation fault.
MON.REX	NO	start senza parametri è indicato come comando errato: manca un messaggio di errore più esaustivo. Se un pid non ha blocchi allocati, lab pid genera l'errore "errore lettura da FIFO in: list bloc k"
MON.AFOK	NO	Strani caratteri mostrati con la lab
MON.CPT	NO	In caso di test il server si pianta e non risponde più al monitor.
MON.BF	SI	
LIB.AFOK	NO	I puntatori riportati sono sbagliati: un blocco da un byte risulta grande 2 byte
LIB.CPT	NO	Errore come in MON.CPT
LIB.CAL	NO	Genera segmentation fault...
LIB.REA	NO	Genera segmentation fault...
		Stile costante e leggibile
		Scarso uso di macro
		Gestione comunicazione molto discutibile
		Include inclusi con path relativo
		Esito system call non controllato
Qualità codice		Errore grave nella gestione della put (get): usata funzione strcat (strcpy) in libsmm.c
Modularità		Discreta
Relazione		Buona la struttura. Manca la descrizione del protocollo di comunicazione.
		Per la libreria è stato necessario rinominare la funzione smm_chk_blk in smm_chk_ptr come da specifiche.
		Inoltre il makefile non prevedeva di inserire il file converti nella libreria, che pertanto non permetteva la compilazione di smm_test.
NOTE		Viene il dubbio se il codice è stato testato con smm_test.