Micro Shell (ush): specifiche per il progetto d'esame e per l'esonero n.2

Laboratorio di Sistemi Operativi I
Anno Accademico 2006-2007

Francesco Pedullà
(Tecnologie Informatiche)

Massimo Verola
(Informatica)

Copyright © 2006-2007 Francesco Pedullà, Massimo Verola
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license can be found at: http://www.gnu.org/licenses/fdl.html#TOC1

Il progetto

- Questo modulo descrive le specifiche del progetto da svolgere per l'esame di Laboratorio di Sistemi Operativi I - AA 2006-2007.
- Tali specifiche si applicano anche al compito di esonero n.2, il quale pero' richiede l'implementazione di un numero inferiore di funzionalita'.
- **NOTA BENE**: il progetto deve costituire ***una creazione originale***, quindi non è possibile condividere parti di codice o della eventuale relazione, se richiesta, con altri studenti/gruppi, o copiare contenuti derivanti da altre fonti.
- **Il progetto richiede lo sviluppo di una shell minimale, denominata ush (micro shell)**

© 2006 F. Pedullà, M. Verola

Il progetto

- La micro shell dovra' emulare il comportamento della *Thompson shell*:

  The Thompson shell was the first Unix shell, introduced in the first version of Unix in 1971, and was written by Ken Thompson. It was a simple command interpreter, not designed for scripting, but nonetheless introduced several innovative features to the command line interface and led to the development of the later Unix shells.

  NAME sh -- shell (command interpreter)
  SYNOPSIS sh [name [arg_1 ... [arg_9]]]

  DESCRIPTION

  sh is the standard command interpreter. It is the program which reads and arranges the execution of the command lines typed by most users. It may itself be called as a command to interpret files of commands. [...]

© 2006 F. Pedullà, M. Verola

Funzionalita' della micro shell `ush` - I

- agisce da interprete interattivo di comandi:
  - presenta all'utente un prompt in attesa di un comando
  - legge l'intera linea di comando ad ogni Enter digitato
  - effettua il parsing della linea ed eventuali elaborazioni sulle stringhe
  - esegue il comando e al suo completamento ritorna a mostrare il prompt
- esegue un file contenente una serie di comandi
- esegue comandi interni e lancia comandi esterni in foreground e background
- supporta le sequenze di comandi
- gestisce gli interrupts provocati da certi caratteri speciali digitati a terminale (**`Ctrl+c`** che genera **`SIGINT`** e **`Ctrl+\`** che genera **`SIGQUIT`**)

© 2006 F. Pedullà, M. Verola

Funzionalita' della micro shell `ush` - II

- Supporta:
  - la redirezione dell'I/O
  - le pipe tra comandi
  - la *filename expansion*
  - il *quoting*
  - Il passaggio di parametri

© 2006 F. Pedullà, M. Verola

Specifiche per l'esonero n. 2

- Le specifiche sono indicate dalla man page della Thompson shell Version 3 (http://www.in-ulm.de/~mascheck/bourne/v3/), disponibile sul sito del Corso, **con le seguenti varianti**:
  - ***Termination Reporting***: ignorare
  - ***Redirection of I/O***: aggiungere il supporto per la redirezione dello *stderr* mediante la notazione **2>** e **2>>**
  - ***Pipes and Filters***: ignorare le specifiche dell'intero paragrafo ed implementare le pipe di shell mediante l'usuale notazione **|** , come nell'esempio:
    `$ coma | comb | comc`
  - ***Quoting***: implementare solo la notazione *single quote* **‘ ‘** e *double quote* **“ “** (ignorare la notazione *backslash* \)

© 2006 F. Pedullà, M. Verola

Specifiche per il progetto (appello normale)

- Oltre alle specifiche richieste per l'esonero 2, la **`ush`** dovra' supportare **2 funzionalita' a scelta** tra le seguenti:
  - ricerca dei comandi esterni mediante la variabile **`PATH`**
  - creazione di variabili ed espansione di variabili mediante la notazione **`$NOME_VARIABILE`**
  - command substitution mediante la notazione **`$(command)`**
  - job control: sospensione di un comando mediante la sequenza **`Ctrl+z`**, riattivazione in foreground o background mediante i comandi *built-in* **`fg %job_`*`num`*** e **`bg %job_num`**, terminazione mediante il comando **`kill %job_num`**

© 2006 F. Pedullà, M. Verola

Il progetto: possibili estensioni

- Si possono sviluppare le seguenti ulteriori funzionalita' alla **ush**, da implementare opzionalmente e solo dopo che tutte le funzionalita' richieste siano state sviluppate correttamente:
  - alias
  - history: richiamo di un comando passato mediante la notazione **!*num_comando*** (ad esempio **!135**, per richiamare ed invocare il comando n. 135 della *history list*)

© 2006 F. Pedullà, M. Verola

Alcune considerazioni e precisazioni - I

- **ush** puo' agire da ***shell interattiva***, se viene invocata senza argomenti, oppure da ***interprete di file di comandi***
- Nella modalita' interattiva, il prompt della **ush** deve avere il seguente formato:
  - pathname assoluto della directory corrente, spazio, carattere **$,** spazio, come nell'esempio:
    /home/user2/projects/ush $
  - deve aggiornarsi ogni volta che si effettua un **cd**
- Nella modalita' interprete di file di comandi, **ush** viene invocata passandole un nome di file che contiene una serie di linee di comando, una per linea, le quali vengono eseguite in sequenza come se fossero digitate al prompt; in questo caso e' possibile passare degli argomenti a cui il file di comandi puo' riferirsi mediante l'usuale notazione **$n** (**n**-esimo argomento)
- Il file di comandi deve essere eseguito anche mediante la digitazione del suo nome ed eventuali argomenti al prompt della shell interattiva

© 2006 F. Pedullà, M. Verola

Alcune considerazioni e precisazioni - II

- Per semplificare il *parsing* della linea di comando, si puo' assumere che tutte le seguenti sequenze di metacaratteri siano separate mediante spazio dagli argomenti adiacenti:
  - **;** utilizzato per la sequenza di comandi
  - **< > >> 2> 2>>** utilizzati nella redirezione dell'I/O
  - **|** utilizzato per le pipe tra comandi
  - **&** utilizzato per l'esecuzione di comandi in background

© 2006 F. Pedullà, M. Verola

Package da consegnare - I

- La compilazione ed installazione del programma **`ush`** deve essere gestita mediante **`Makefile`** e **NON** deve richiedere il privilegio di root
- Il progetto deve essere documentato mediante una breve relazione che descrive:
  - l'organizzazione del package
  - la struttura del programma
  - le scelte implementative
  - il dettaglio delle funzionalita' supportate (con precisazioni sulle eventuali limitazioni)
  - le informazioni necessarie alla corretta compilazione, installazione e test (variabili d'ambiente, file di configurazione, ...)
  - eventuali file di comandi per il test delle varie funzionalita'

© 2006 F. Pedullà, M. Verola

Package da consegnare - II

- La consegna deve consistere in un package, da spedire via mail, in formato tar compresso (**`tar czf`**) col nome *ush.tgz,* contenente i sorgenti C, gli include file, il Makefile, la documentazione ed eventuali file per il test

© 2006 F. Pedullà, M. Verola