הפעלה-

* חלק יבש-
  + אביטל – שאלה 1
  + אלון – שאלה 2

משימות

* ~~להרים גיט~~
* להוריד ויז'ואל סטודיו בVM
  + לעבוד יחד בלייב שייר?
  + להחליט איך נוח
  + לחבר גיט לIDE בVM, כדי שנוכל לעבוד יחד על ה-VM
* לחלק משימות
  + לדבר עם רועי ועידן איך נכון לגשת
* משימות
  + ללכת לפי הסדר שב-README
  + אביטל
    - פונקציות של הJOBSLIST (כמו ADD וכו')
    - EXTERNAL
    - בודק שרץ ועושה בנאים וכו'
  + אלון
    - SPECIAL
      * לבדוק אם יש שגיאות שצריך לטפל בהן בPIPES
        + התאמות ל-TIMEOUT
      * Waitwrapper?
* טסטים
  + למצוא משהו כללי
  + לכתוב
  + טסטים שצריך-
    - לוודא שהפקודות עובדות
      * נראה שבינתיים חלק מהBUILTIN עובדות
  + לבדוק שתהליכים לא נשארים שלא עשינו לו WAIT (כמו שלא עושים FREE)
* לכתוב הערות
* לוודא שמות שונים מפוקנציות של עידן וסארי וכו'
* לבדוק WARNINGS וכו'

Cd – should be able to get /home/XXXX?

ctrlZ – should not get us out of smash?

Idan’s-

PipedExecute

ActivatePiped

HasPipe

ContainSpecialCommand

SplitRedirection

splitPipe

Where to modify things in code for piping:

* Execute
  + If pipe – execute two commands, deal with pipes (open and close fd’s)

RedirectionCommand

RedirectionCommand::execute

PipeCommand

PipeCommand::execute

PipedExecute

ActivatePipe

checkSpecialType

SplitPipeCommand

One for piped

One for redirection (returns the file to input as a second command)

Check if “command |command” is ok

Pipe-

1. Open a pipe
2. Child 1- sender
   1. Close pipe[write]
   2. Redirect pipe[read] into STDOUT
   3. Execute command
   4. Close pipe[read]
3. Childe 2- receiver
   1. Close pipe[read]
   2. Redirect pipe[write] into STDIN
   3. Execute command
   4. Close pipe[write]
4. Father
   1. Wait for child processes?
   2. Close pipe[read] and pipe[write]

Delete and new for commands

bool PipeCommand::storeFDs(){

org\_fds[0] = dup(STDIN\_FILENO);

org\_fds[1] = dup(STDOUT\_FILENO);

org\_fds[2] = dup(STDERR\_FILENO);

for(int i = 0; i < 3; ++i){

if(org\_fds[i] == -1){

perror("smash error: dup failed");

return false;

}

}

return true;

}

bool PipeCommand::restoreFDs(){

dup2\_in = dup2(org\_fds[0], STDIN\_FILENO);

dup2\_out = dup2(org\_fds[1], STDPOUT\_FILENO);

dup2\_err = dup2(org\_fds[2], STDERR\_FILENO);

if(dup2\_in == -1 || dup2\_out == -1 || dup2\_err == -1){

perror("smash error: dup2 failed");

return flase

}

return true;

}