42sh: mini-shell

Extra tips en aanwijzingen

Over de shell zelf

- je shell hoeft *niet* de volgende features implementeren:
  - bestandnamen automatisch invullen als de characters "`*`" of "?" worden gebruikt;
  - de thuismap van de gebruiker invullen als de character "~" wordt gebruikt.
- als je niet precies weet waar je moet beginnen, focus dan eerst op "simpele" commando's. Dit is knoptype "`COMMAND`". Dus in `shell.c`, `run_command`:

```
if (node->type == NODE_COMMAND)
{
    char *program = node->command.program;
    char **argv = node->command.argv;
    // hier een goede combinatie van fork + exec
    ...
}
```

- een "gewone" POSIX shell ondersteunt redirecties op alle plekken in een eenvoudige commando: "`ls >foo`" en "`>foo ls`" zijn equivalent. **Het gegeven voorbeeld code kent alleen de 2de vorm:** "`>foo ls`"
- **in een "pipe" constructie moeten alle onderdelen ge``fork``t worden**, zelfs als ze alleen "built-in" commando's bevatten. Dit houdt de implementatie eenvoudiger. Bijvoorbeeld:

```
exit 42 # eindigt de shell
exit 42 | sleep 1  # exit in sub-shell, hoofdshell blijft draaien

cd /tmp # verandert de huidige map
cd /tmp | sleep 1  # cd in sub-shell, hoofdshell blijft in zelfde map
```

- `valgrind` zegt "still reachable: NNN in NNN blocks". Het lijkt erop dat de basiscode al geheugen lekt. We maken echter een onderscheid tussen geheugen gelekt door jullie code, en geheugen gelekt door de onze. Om zelf dit te onderscheiden, draai `valgrind` dan met:

```
valgrind --leak-check=full --show-reachable=yes ...
```

  Alles gelekt door `rl_initialize`, `read_history`, `history_search`, `add_history` en `reset_text` is buiten jullie controle en wordt dus niet beoordeeld als fout.

Over de omgeving

- indien geen `clang` aanwezig is op het systeem: `make CC=gcc ...`
- indien OSX: `make LIBS=-lreadline ...`
- indien OSX: `valgrind --dsymutil=yes ...`
- eigen privé Git repository bij de UvA: https://gitlab-fnwi.uva.nl/